On Transitive Consistency for Linear Invertible Transformations between Euclidean Coordinate Systems ⋆

Johan Thunberg [a] , Florian Bernard [a,b], Jorge Goncalves [a,c]

[a] *Luxembourg Centre for Systems Biomedicine, University of Luxembourg, Esch-sur-Alzette, LUXEMBOURG*

[b] *Centre Hospitalier de Luxembourg, Luxembourg City, LUXEMBOURG*

[c] *Control Group, Department of Engineering, University of Cambridge, Cambridge, UNITED KINGDOM*

Abstract

Transitive consistency is an intrinsic property for collections of linear invertible transformations between Euclidean coordinate frames. In practice, when the transformations are estimated from data, this property is lacking. This work addresses the problem of synchronizing transformations that are not transitively consistent. Once the transformations have been synchronized, they satisfy the transitive consistency condition – a transformation from frame A to frame C is equal to the composite transformation of first transforming A to B and then transforming B to C. The coordinate frames correspond to nodes in a graph and the transformations correspond to edges in the same graph. Two direct or centralized synchronization methods are presented for different graph topologies; the first one for quasi-strongly connected graphs, and the second one for connected graphs. As an extension of the second method, an iterative Gauss-Newton method is presented, which is later adapted to the case of affine and Euclidean transformations. Two distributed synchronization methods are also presented for orthogonal matrices, which can be seen as distributed versions of the two direct or centralized methods; they are similar in nature to standard consensus protocols used for distributed averaging. When the transformations are orthogonal matrices, a bound on the optimality gap can be computed. Simulations show that the gap is almost tight, even for noise large in magnitude. This work also contributes on a theoretical level by providing linear algebraic relationships for transitively consistent transformations. One of the benefits of the proposed methods is their simplicity – basic linear algebraic methods are used, e.g., the Singular Value Decomposition (SVD). For a wide range of parameter settings, the methods are numerically validated.

Key words: Distributed optimization, transformation synchronization, Procrustes problem, consensus algorithms, graph theory.

1 Introduction

Collections of linear invertible transformations between Euclidean coordinate systems must be transitively consistent. In practice however, when the transformations are estimated from data, this condition does not hold. This issue is present in the 3D localization problem, where transformations are rigid and estimated from e.g., camera measurements; in the multiple images registration problem where the transformations are affine (or linear by using homogeneous coordinates); in the generalized Procrustes problem where scales, rotations and translations are calculated from multiple point clouds. In order to resolve the issue, the estimated transformations need to be synchronized in the sense of finding transitively consistent transformations close to the estimated ones.

1.1 Problem

This work addresses the problem of synchronizing linear invertible transformations or matrices between Euclidean coordinate systems or frames. More precisely, given a collection $\{G_{ij}\}$ of matrices in $GL(d,\mathbb{R})$, another collection $\{G^*_{ij}\}$ of matrices in $GL(d,\mathbb{R})$ is constructed such that

$$G^*_{ij}G^*_{jk} = G^*_{ik}, \text{ for all } i,j,k, \tag{1}$$

⋆ The authors gratefully acknowledge the financial support from the Fonds National de la Recherche, Luxembourg (6538106, 8864515).

Email addresses: johan.thunberg@uni.lu (Johan Thunberg), bernard.florian@chl.lu (Florian Bernard), jorge.goncalves@uni.lu (Jorge Goncalves).

where G_{ij}^* is "close" to G_{ij} for all i, j. By satisfying (1), the collection $\{G_{ij}^*\}$ is said to be transitively consistent.

1.2 Background

There are many applications for the proposed methods. One such application is the 3D localization problem in camera networks [1] where a network of cameras are observing a scene and epipolar geometry is used to calculate/measure G_{ij} transformations between (i, j)-pairs of cameras. If the cameras are fully calibrated these transformations are Euclidean, otherwise they could be e.g., affine (or linear by using homogeneous coordinates). Since the transformations are calculated from measurements, they do not satisfy (1) in general. Hence our proposed methods can be used to synchronize the matrices. For the 3D localization problem, we do not have to limit ourselves to the case of cameras and epipolar geometry. The transformations could be calculated in a setting where the geometry of the scene is known. In the case of known point features, the perspective-n-point problem can be solved in order to get estimates of the relative transformations [2].

Another important problem is image registration, which has attracted much attention in the medical imaging community. The number of applications is vast, ranging from surgery planning to longitudinal studies. To register a (moving) image with another (fixed) image is to transform the former into the the latter in such way that they fit in the "best" way. For that, optimization methods are used to calculate a transformation which minimizes a suitable objective. Registration of multiple images poses a greater challenge. There are several approaches in the literature. For example: Finding a path of pairwise transformations, which contains all images [3]; aligning images with a reference frame [4]; image congealing, where variability along known axes of variation is removed in an iterative manner [4]; considering a minimum description length (MDL) approach of a statistical shape model built from the correspondences given due to groupwise image registration [5]; Bayesian formulations and Expected Maximization (EM) [6].

Another way to solve the (affine) multiple images registration problem is to use the transitive consistency criterion (1) [7]. Let the G_{ij} correspond to the affine transformations calculated from pairwise registrations, then our method can be used to create transitively consistent G_{ij}^* transformations. Registration methods using transitive consistency have also been proposed for deformable transformations [8,9].

A related problem to the one posed in this paper is the problem of calculating the "best" translations, rotations and scales between pairs of point clouds. If only one pair is considered the problem is referred to as the Procrustes Problem [10]. This problem can be solved by means of singular value decomposition or eigenvalue decomposition [11,12,13], or in the case case of 3D transformations, by a quaternion-based approach [14,15]. The problem restricted to 3D is referred to as the absolute orientation problem [13,14]. In the general setting, when n point clouds are considered, the problem is referred to as the Generalized Procrustes Problem [10]. In order to solve this problem, iterative methods are often used; when the dimension is two or three, direct methods have recently been proposed [16]. Our previous work in [17] has tackled the Generalized Procrustes Problem using an approach based on transitive consistency. The present paper will extend and generalize these ideas as well as describe many theoretical properties of the generalizations.

Our methods can be used for solving the Generalized Procrustes Problem in the following way: Between each pair of point clouds a G_{ij} transformation is calculated using any standard technique [11,12,13], then our methods are used to improve the pairwise transformations by calculating transitively consistent transformations.

In the special case when the G_{ij} are orthogonal matrices, Singer *et al.* have presented methods for the optimization of transitive consistency [18,19,20,21]. These works were later adapted by Pachauri *et al.* to the special case when the G_{ij} are permutation matrices [22]. In the latter work, a relaxation of the original problem is considered – in the original problem the transformations shall be orthogonal matrices – and then permutation matrices are obtained by means of projection from the solution of the relaxed problem. The method presented by Singer *et al.* is said to be a synchronization method for minimization of transitive consistency errors – a formalism adopted in this work.

1.3 Methods and results

The approach in this work share similarities with the approaches of Singer *et al.* and Pachauri *et al.*; it continues along the lines of the the recently proposed methods in [17,23].

In [17,23] a so called Z-matrix is constructed from the G_{ij} matrices. If the index set for the (available) transformations has a certain property, transitively consistent transformations can be obtained by a method where the Singular Value Decomposition (SVD) is calculated for Z. The property that must be fulfilled for the index set $\{(i, j)\} = \mathcal{E}$, is that it is the edge set of a quasi-strongly connected (QSC) directed graph (see Definition 2). In the Z-matrix approach, a set of linear algebraic equations are formulated – equations which shall be satisfied for the case of transitively consistent transformations. When the transformations are not transitively consistent, the problem is solved in the sense of least squares minimization.

As we will show in this work, the Z-matrix appears in the construction of a Hessian matrix H for a quadratic convex function of the G_{ij}, and under certain conditions it holds that $H = Z + Z^T$. From the SVD of the Hessian matrix H, transitively consistent transformations can be calculated in the same manner as for the Z-matrix. The justification for using the H-matrix stems from the fact that it is the Hessian matrix of the objective function in a relevant optimization problem. The justification of using the Z-matrix stems purely from the linear algebraic constraints that should be satisfied for transitively consistent transformations.

The Z-matrix method and the H-matrix method are both direct methods, i.e., the solution is found at once. As an extension we also propose an iterative Gauss-Newton method, which uses the solution from the H-matrix method as initialization. For orthogonal matrices one can prove that this iterative scheme cannot decrease the objective function at all. The Gauss-Newton method is also adapted to the cases of affine and Euclidean transformations. In this case – as opposed to the result for orthogonal matrices – significant improvement over the Z-matrix method and the H-matrix method can be seen in numerical simulations.

Many properties of the Z-matrix and the H-matrix are proved in this work. For example it is shown that transitive consistency in the case of connected graphs is equivalent to the condition that the nullspace of H has dimension d. Furthermore, the transitively consistent transformations can be obtained as the $d \times d$ blocks in a matrix, the columns of which span the nullspace of H. For the Z-matrix only a weaker condition is formulated; if the graph is QSC and the transformations are transitively consistent, the transformations can be obtained as the $d \times d$ blocks in a matrix, the columns of which span the nullspace of Z.

Now, in most aspects the H-matrix approach seems to be superior to the Z-matrix approach. However, one large benefit of using the Z-matrix over the H-matrix is that it can be used in a distributed algorithm when the communication graph is directed.

In a later part of the paper, two distributed methods are introduced for the case of orthogonal G_{ij} transformations. The first method is using the Z-matrix under the assumption that the communication graph is directed and QSC. The other method is using the H-matrix under the assumption that the communication graph is symmetric. The performance of the two methods are almost the same in numerical experiments. The distributed methods are similar in structure to linear consensus protocols [24,25,26,27,28]. Key differences to those approaches is that the states here are matrices instead of vectors, and the states combined converge to a d-dimensional linear subspace instead of the consensus set.

The distributed iterative methods are introduced mainly with communication between agents in mind, e.g., in networks of robots with limited communication range, where the robots only communicate with their neighbors (directly or indirectly). However, a further scenario of the distributed methods is parallelisation in order to better deal with the computational burden in the case of very large problem instances.

When it is known that the transitively consistent transformations are orthogonal matrices, i.e., elements of $O(d) = \{R : R \in \mathbb{R}^{d\times d}, R^T R = I\}$, a method is provided for calculating an upper bound on the optimality gap. In the case when the G_{ij} are also orthogonal, simulations show that this gap is almost tight. As an example, for $n = 100$ coordinate systems, dimension $d = 3$, and randomly generated G_{ij} matrices in $O(3)$, the gap is smaller than a tenth of a percent in average. There are (and will be even more in the future) applications where large networks of cameras, robots, satellites or unmanned vehicles, need to synchronize their pairwise relative rotations. In such applications methods that are near optimal and run almost in real time are of utmost importance to have.

1.4 Outline

The paper proceeds as follows. In Section 2, graphs and properties thereof are introduced, followed by the introduction of the G_{ij} transformations and their connections to the graphs. We have chosen to incorporate graphs in the very definition of transitive consistency. Section 3 addresses linear invertible transformations. In Section 3.1, the Z-matrix is introduced, followed by a collection of results and a least squares method. In Section 3.4, the H-matrix is introduced; in the same manner as in Section 3.1, a collection of results is provided in conjunction with an algorithm. In Section 3.7 a Gauss-Newton method is presented, where the matrices obtained from the H-matrix method are used as initialization. Section 4 consider the special case of orthogonal matrices, i.e., elements of $O(d)$. The section starts with some bounds on the optimality gap, and continues in Section 4.1 with the introduction of distributed algorithms. The reader interested in the distributed methods can go directly to this section and consult the earlier sections only for reference. Section 4.2 is a small detour, where a gradient flow method is presented for orthogonal matrices. This method is employed as a baseline method, used for comparison in some of the simulations in Section 5 – the section where the proposed methods are thoroughly numerically evaluated.

2 Preliminaries

2.1 Directed Graphs

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the node set and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the edge set. The set $\mathcal{N}_i$ is defined by

$$\mathcal{N}_i = \{j : (i, j) \in \mathcal{E}\}.$$

The adjacency matrix $A = [A_{ij}]$ for the graph $\mathcal{G}$ is defined by

$$A_{ij} = \begin{cases} 1 & \text{if } (i, j) \in \mathcal{E}, \\ 0 & \text{else.} \end{cases}$$

The graph Laplacian matrix is defined by

$$L = \text{diag}(A1_n) - A,$$

where $1_n \in \mathbb{R}^n$ is a vector with all entries equal to 1. In order to emphasize that the adjacency matrix A, the Laplacian matrix L and the $\mathcal{N}_i$ sets depend on the graph $\mathcal{G}$, we may write $A(\mathcal{G})$, $L(\mathcal{G})$ and $\mathcal{N}_i(\mathcal{G})$ respectively. For simplicity however, we mostly omit this notation and simply write A, L, and $\mathcal{N}_i$.

Definition 1 (connected graph, undirected path)
The directed graph $\mathcal{G}$ is connected if there is an undirected path from any node in the graph to any other node. An undirected path is defined as a (finite) sequence of unique nodes such that for any pair (i, j) of consecutive nodes in the sequence it holds that

$$((i, j) \in \mathcal{E}) \text{ or } ((j, i) \in \mathcal{E}).$$

Definition 2 (quasi-strongly connected graph, center, directed path)
The directed graph $\mathcal{G}$ is quasi-strongly connected (QSC) if it contains a center. A center is a node in the graph to which there is a directed path from any other node in the graph. A directed path is defined as a (finite) sequence of unique nodes such that any pair of consecutive nodes in the sequence comprises an edge in $\mathcal{E}$.

Definition 3 (symmetric graph)
The directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is symmetric if

$$((i, j) \in \mathcal{E}) \Rightarrow ((j, i) \in \mathcal{E}) \text{ for all } (i, j) \in \mathcal{V} \times \mathcal{V}.$$

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the graph $\bar{\mathcal{G}} = (\mathcal{V}, \bar{\mathcal{E}})$ is the graph constructed by reversing the direction of the edges in $\mathcal{E}$, i.e., $(i, j) \in \bar{\mathcal{E}}$ if and only if $(j, i) \in \mathcal{E}$. It is easy to see that

$$A(\bar{\mathcal{G}}) = (A(\mathcal{G}))^T \text{ and } L(\bar{\mathcal{G}}) = \text{diag}((A(\mathcal{G}))^T 1_d) - A(\mathcal{G})^T.$$

2.2 Transformations

Given a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, let there be a collection of matrices $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ where $G_{ij} \in GL(d, \mathbb{R})$ for all $(i, j) \in \mathcal{E}$. Let $n = |\mathcal{V}|$. The G_{ij} are not necessarily transitively consistent in that

$$G_{ik} \neq G_{ij} G_{jk}$$

may hold if $(i, j), (j, k)$ and (i, k) are elements of $\mathcal{E}$.

In the methods to be defined, the goal is to find a transitively consistent collection $\{G^*_{ij}\}_{(i,j)\in\mathcal{E}}$ of matrices in $GL(d, \mathbb{R})$, such that for all $(i, j) \in \mathcal{E}$, G^*_{ij} is close to G_{ij} in some appropriate sense. Notation-wise, G^*_{ij} is simply (a name of) a matrix. This notation should not be mixed up with the conjugate transpose – in this paper, all matrices considered are real and the conjugate transpose will not be used.

Definition 4 (transitive consistency)

(1) The matrices in the collection $\{G^*_{ij}\}_{(i,j)\in\mathcal{V}\times\mathcal{V}}$ of matrices in $GL(d, \mathbb{R})$ are transitively consistent for the complete graph *if*

$$G^*_{ik} = G^*_{ij} G^*_{jk}$$

for all i, j and k.

(2) Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the matrices in the collection $\{G^*_{ij}\}_{(i,j)\in\mathcal{E}}$ of matrices in $GL(d, \mathbb{R})$ are transitively consistent for $\mathcal{G}$ *if there is a collection $\{G^*_{ij}\}_{(i,j)\in\mathcal{V}\times\mathcal{V}} \supset \{G^*_{ij}\}_{(i,j)\in\mathcal{E}}$ such that $\{G^*_{ij}\}_{(i,j)\in\mathcal{V}\times\mathcal{V}}$ is transitively consistent for the complete graph.*

If it is apparent by the context, sometimes we will be less strict and omit to mention which graph a collection of transformations is transitively consistent for. A sufficient condition for transitive consistency of the G^*_{ij} matrices for any graph is that there is a collection $\{G^*_i\}_{i\in\mathcal{V}}$ of matrices in $GL(d, \mathbb{R})$ such that

$$G^*_{ij} = G^{*-1}_i G^*_j$$

for all i, j. Lemma 6 below and the proof thereof provides additional important information. The result is similar to that in [1]. For the statement of the lemma, the following definition is needed.

Definition 5 *Two collections $\{G^*_i\}_{i\in\mathcal{V}}$ and $\{G^{**}_i\}_{i\in\mathcal{V}}$ of matrices in $GL(d, \mathbb{R})$ are equal up to transformation from the left, if there is $Q \in GL(d, \mathbb{R})$ such that*

$$QG^*_i = G^{**}_i \text{ for all } i.$$

Lemma 6 *For any graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and collection $\{G^*_{ij}\}_{(i,j)\in\mathcal{E}}$ of matrices in $GL(d,\mathbb{R})$ that are transitively consistent for $\mathcal{G}$,*

(1) there is a collection $\{G^*_i\}_{i\in\mathcal{V}}$ of matrices in $GL(d,\mathbb{R})$ such that

$$G^*_{ij} = G^{*-1}_i G^*_j \text{ for all } (i,j) \in \mathcal{E}, \qquad (2)$$

(2) all collections $\{G^*_i\}_{i\in\mathcal{V}}$ satisfying (2) are equal up to transformation from the left if and only if $\mathcal{G}$ is connected,

(3) there is a unique collection $\{G^*_{ij}\}_{(i,j)\in\mathcal{V}\times\mathcal{V}} \supset \{G^*_{ij}\}_{(i,j)\in\mathcal{E}}$ of transitively consistent matrices for the complete graph, if and only if all collections $\{G^*_i\}_{i\in\mathcal{V}}$ satisfying (2) are equal up to transformation from the left.

Proof: All matrices appearing in this proof, if the contrary is not explicitly stated, are assumed to be elements of $GL(d,\mathbb{R})$.

(1) Since the matrices in $\{G^*_{ij}\}_{(i,j)\in\mathcal{E}}$ are transitively consistent for $\mathcal{G}$, there is $\{G^*_{ij}\}_{(i,j)\in\mathcal{V}\times\mathcal{V}} \supset \{G^*_{ij}\}_{(i,j)\in\mathcal{E}}$ in which the matrices are transitively consistent for the complete graph. Let the G^*_i in a collection $\{G^*_i\}_{i\in\mathcal{V}}$ be defined by

$$G^*_i = G^*_{1i}.$$

We shall prove that

$$G^{*-1}_i G^*_j = G^{*-1}_{1i} G^*_{1j} = G^*_{ij} \text{ for all } i,j.$$

Using the fact that G^*_{11} is invertible and the fact that $G^*_{11} = G^{*2}_{11}$, one can show that $G^{*2}_{11} = I$. Now, $G^*_{1i}G^*_{i1} = G_{11} = I$; thus $G^{*-1}_{1i} = G^*_{i1}$. But then

$$G^{*-1}_{1i} G^*_{1j} = G^*_{i1} G^*_{1j} = G^*_{ij}.$$

(2) We know that transitive consistency of $\{G^*_{ij}\}_{(i,j)\in\mathcal{E}}$ for $\mathcal{G}$ is equivalent to the statement that there is a collection $\{G^*_i\}_{i\in\mathcal{V}}$ of matrices in $GL(d,\mathbb{R})$ such that

$$G^*_{ij} = G^{*-1}_i G^*_j \text{ for all } (i,j) \in \mathcal{E}.$$

Let $G^*_{ij} = G^{-1}_i G_i$ for all $i,j \in \mathcal{V}$. For any other collection $\{G^{**}_i\}_{i\in\mathcal{V}}$ of matrices in $GL(d,\mathbb{R})$ such that

$$G^*_{ij} = G^{**-1}_i G^{**}_j \text{ for all } (i,j) \in \mathcal{E},$$

it holds that

$$\begin{aligned} &\left[G^{**T}_1 \; G^{**T}_2 \; \dots \; G^{**T}_n\right]^T \\ &= \text{diag}\,(Q_1, Q_2, \dots, Q_n) \cdot \left[G^{*T}_1 \; G^{*T}_2 \; \dots \; G^{*T}_n\right]^T, \end{aligned}$$

where the Q_i matrices are elements of $GL(d,\mathbb{R})$.

If: Now, if the graph is connected and at least two of the Q_i are not equal, there is $(j,k) \in \mathcal{E}$ such that $Q_j \neq Q_k$. We know

$$G^*_j G^*_{jk} G^{*-1}_k = I_d,$$

but since $Q_j \neq Q_k$ we can calculate this entity to

$$G^*_j G^*_{jk} G^{*-1}_k = Q^{-1}_j Q_k \neq I_d,$$

which is a contradiction. $I_d \in \mathbb{R}^{n\times n}$ is the identity matrix.

Only if: On the other hand, if the graph is not connected there are two disjoint sets $\mathcal{V}_1$ and $\mathcal{V}_2$ such that $\mathcal{V}_1 \cup \mathcal{V}_2 = \mathcal{V}$, for which there is no pair $(i,j) \in \mathcal{E}$ such that ($i \in \mathcal{V}_1$ and $j \in \mathcal{V}_2$) or ($j \in \mathcal{V}_1$ and $i \in \mathcal{V}_2$). Thus, the nodes in $\mathcal{V}_1$ and the corresponding edges, respective the nodes in $\mathcal{V}_2$ and the corresponding edges, can be seen as two different disconnected (sub)graphs, each of them being connected; the G^*_i matrices in the first graph can be multiplied with a matrix Q_1 from the left and the G^*_i matrices in the second graph can be multiplied with a matrix Q_2 from the left, where $Q_1 \neq Q_2$, generating a collection of matrices $\{G^{**}_i\}_{i\in\mathcal{V}}$ not equal to $\{G^*_i\}_{i\in\mathcal{V}}$ up to transformation from the left.

(3) **If:** Any other collection $\{G^{**}_{i\in\mathcal{V}}\}$ of matrices in $GL(d,\mathbb{R})$ such that

$$G^*_{ij} = G^{**-1}_i G^{**}_j \text{ for all } (i,j) \in \mathcal{E},$$

is equal to $\{G^*_i\}$ up to transformation from the left. Now, for any $(i,j) \in (\mathcal{V}\times\mathcal{V}) - \mathcal{E}$ it holds that

$$G^{**-1}_i G^{**}_j = G^{*-1}_i Q^{-1} Q G^*_j = G^{*-1}_i G^*_j = G^*_{ij}$$

for some matrix $Q \in GL(d,\mathbb{R})$.

Only if: The approach here is similar to that in 2) above. Suppose for $\{G^*_{i\in\mathcal{V}}\}$ satisfying (2), there is another collection $\{G^{**}_{i\in\mathcal{V}}\}$ of matrices in $GL(d,\mathbb{R})$ also satisfying (2), but the matrices in the two collections are not equal up to transformation from the left. Then it holds that

$$\begin{aligned} &\left[G^{**T}_1 \; G^{**T}_2 \; \dots \; G^{**T}_n\right]^T \\ &= \text{diag}\,(Q_1, Q_2, \dots, Q_n) \cdot \left[G^{*T}_1 \; G^{*T}_2 \; \dots \; G^{*T}_n\right]^T, \end{aligned}$$

where the Q_i matrices are elements of $GL(d,\mathbb{R})$ and there is a pair (k,l) for which $Q_k \neq Q_l$.

Now

$$G^{**-1}_k G^{**}_l = G^{*-1}_k Q^{-1}_k Q_l G^*_l \neq G^{*-1}_k G^*_l.$$

■

Lemma 6 states that connectivity is a necessary property to determine a unique (up to transformation from the left) collection $\{G_i^*\}$ satisfying (2). As it turns out, a stronger type of connectivity – quasi-strong connectivity – is useful in order to develop linear algebraic methods for solving our synchronization problem. The first method we present is based on the so called Z-matrix.

3 Linear invertible transformations

3.1 The Z-matrix

In this section a certain matrix is defined – referred to as Z. It is used as a building block in a matrix H, corresponding to the Hessian of a convex quadratic function, see Section 3.4. After its definition, its properties are investigated. Amongst other things, it is shown that if the G_{ij} transformations are orthogonal, i.e., $G_{ij}^T G_{ij} = I$, the matrix $(-Z)$ is (critically) stable in the linear dynamical systems sense (cf. Lemma 14). This means that, for directed graphs, the matrix Z can be used in a linear distributed algorithm for synchronizing orthogonal matrices (Section 4.1).

Define the matrix

$$W(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}}) = [W_{ij}(G_{ij}^*)],$$

where

$$W_{ij}(G_{ij}^*) = \begin{cases} G_{ij}^* & \text{if } j \in \mathcal{N}_i, \\ 0 & \text{else,} \end{cases}$$

and the matrix

$$\begin{aligned} &Z(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}}) \\ &= \text{diag}(A(\mathcal{G})1) \otimes I_d - W(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}}). \end{aligned}$$

The symbol $\otimes$ denotes the Kronecker product.

Remark 7 *A more general way of constructing the W-matrix and the Z-matrix with positive weights is as follows. Replace the W_{ij} in the definition of W with $a_{ij}W_{ij}$, and replace $diag(A(\mathcal{G})1) \otimes I_d$ in the definition of Z with*

$$diag(\sum_{j\in\mathcal{N}_1} a_{1j}, \sum_{j\in\mathcal{N}_2} a_{2j}, \ldots, \sum_{j\in\mathcal{N}_n} a_{nj}) \otimes I_d.$$

The a_{ij} are positive for all i, j. Equivalent results to all the results obtained for the Z-matrix in this section can also be formulated for the alternative Z-matrix with positive weights. The alternative Z-matrix can be used in a distributed algorithm, equivalent to the one that will be presented in Section 4.1.1.

For the collection $\{G_i^*\}_{i\in\mathcal{V}}$ of matrices in $GL(d, \mathbb{R})$, let

$$\begin{aligned} U_1(\{G_i^*\}_{i\in\mathcal{V}}) &= \begin{bmatrix} G_1^{*-T} & G_2^{*-T} & \ldots & G_n^{*-T} \end{bmatrix}^T, \\ U_2(\{G_i^*\}_{i\in\mathcal{V}}) &= \begin{bmatrix} G_1^* & G_2^* & \ldots & G_n^* \end{bmatrix}. \end{aligned}$$

Lemma 8 *For any **(QSC)** graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, collection $\{G_{ij}^*\}_{(i,j)\in\mathcal{E}}$ of matrices in $GL(d, \mathbb{R})$ – transitively consistent for $\mathcal{G}$ – and collection $\{G_i^*\}_{i\in\mathcal{V}}$ of matrices in $GL(d, \mathbb{R})$ it holds that*

$$G_{ij}^* = G_i^{*-1} G_j^* \text{ for all } (i, j) \in \mathcal{E}$$

***(if and)** only if*

$$im(diag\,(G_1^*, G_2^*, \ldots, G_n^*)\, V) = ker(L \otimes I_d), \qquad (3)$$

for any matrix V, where the columns thereof form a basis for $ker(Z(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}}))$. In particular, if $\mathcal{G}$ is QSC, (3) can be stated as

$$im(U_1(\{G_i^*\}_{i\in\mathcal{V}})) = ker(Z(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}})).$$

Proof:
Only if: Suppose it holds that

$$G_{ij}^* = G_i^{*-1} G_j^* \text{ for all } (i, j) \in \mathcal{E}.$$

Then

$$\begin{aligned} &Z(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}}) \\ =\ &\text{diag}(G_1^{*-1}, G_2^{*-1}, \ldots, G_n^{*-1})(L \otimes I)\cdot \\ &\text{diag}(G_1^*, G_2^*, \ldots, G_n^*). \end{aligned} \qquad (4)$$

Now,

$$\begin{aligned} &Z(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}})V = 0 \Leftrightarrow \\ &(L \otimes I)\text{diag}(G_1^*, G_2^*, \ldots, G_n^*)V = 0 \Leftrightarrow \\ &\text{im}(\text{diag}\,(G_1^*, G_2^*, \ldots, G_n^*)\, V) = \text{ker}(L \otimes I_d). \end{aligned}$$

If: This part is only proven for the case when the graph $\mathcal{G}$ is QSC.

Since $\{G_{ij}^*\}_{(i,j)\in\mathcal{E}}$ is transitively consistent for $\mathcal{G}$, there is $\{G_i^{**}\}_{i\in\mathcal{V}}$ of matrices in $GL(d, \mathbb{R})$ such that

$$\begin{aligned} &Z(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}}) \\ =\ &\text{diag}(G_1^{**-1}, G_2^{**-1}, \ldots, G_n^{**-1})(L \otimes I)\cdot \\ &\text{diag}(G_1^{**}, G_2^{**}, \ldots, G_n^{**}). \end{aligned}$$

Thus, the null-space of $Z(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}})$ is given by

$$\text{ker}(Z(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}})) = \text{im}(V),$$

where

$$V = \mathrm{diag}(G_1^{-1**}, G_2^{-1**}, \ldots, G_n^{-1**})([1,1,\ldots,1]^T \otimes I_d).$$

Now, suppose (3) holds. Then

$$\mathrm{diag}\,(G_1^*, G_2^*, \ldots, G_n^*)\, V = ([1,1,\ldots,1]^T \otimes I_d)Q,$$

where Q is some matrix in $GL(d,\mathbb{R})$. This means that

$$\mathrm{diag}\left(G_1^* G_1^{-1**}, G_2^* G_2^{-1**}, \ldots, G_n^* G_n^{-1**}\right) = I_n \otimes Q,$$

which implies that $\{G_i^{**}\}_{i\in\mathcal{V}}$ and $\{G_i^*\}_{i\in\mathcal{V}}$ are equal up to transformation from the left. By using Lemma 6 we can conclude that

$$G_{ij}^* = G_i^{*-1} G_j^* \text{ for all } (i,j) \in \mathcal{E}.$$

■

Remark 9 *In Lemma 8, the relation*

$$im(diag\,(G_1^*, G_2^*, \ldots, G_n^*)\, V) = ker(L \otimes I_d)$$

holds if and only if for any matrix V_2, where the columns thereof comprise a basis for $ker(L \otimes I_d)$, there is a matrix Q such that

$$diag\,(G_1^*, G_2^*, \ldots, G_n^*)\, V = V_2 Q.$$

Remark 10 *In Lemma 8, if $\mathcal{G}$ is connected but not QSC, it can hold that A^T is the adjacency matrix of a QSC graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$. Then it holds that*

$$im(U_1(\{G_i^{*-T}\}_{i\in\mathcal{V}}) = ker(Z(\mathcal{G}', \{G_{ij}^{*T}\}_{(i,j)\in\mathcal{E}}).$$

Lemma 8 is important as it provides a way of finding matrices $\{G_i^*\}_{i\in\mathcal{V}}$ fulfilling (2). In the following subsection, this lemma is used to provide a least squares method.

3.2 A least squares method

Suppose the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is QSC, and the collection $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ of matrices in $GL(d,\mathbb{R})$ are not transitively consistent for $\mathcal{G}$, but close to being transitively consistent (closeness is in the sense of some matrix norm in $\mathbb{R}^{d\times d}$). Then, motivated by Lemma 8, the collection $\{G_i^*\}_{i\in\mathcal{V}}$ of matrices in $GL(d,\mathbb{R})$ such that (2) holds can be found by using the following approach.

Algorithm 1

(1) Solve the problem

$$\min_V \|Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})V\|_F^2,$$

where $V \in \mathbb{R}^{nd\times d}$, $V^TV = I_d$. This is done by means of the Singular Value Decomposition of $Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})$. Let V_1 be the optimal solution.

(2) Identify the G_i^* in the collection $\{G_i^*\}_{i\in\mathcal{V}}$ by

$$V_1^T = [G_1^{*-T}, G_2^{*-T}, \ldots, G_n^{*-T}].$$

The algorithm is motivated by Lemma 8. Note that the method is applicable if and only if the graph $\mathcal{G}$ is QSC, (Lemma 8). In the special case when the transformations are known to be Euclidean (or belong to some other desirable subset of $GL(d,\mathbb{R})$), the collection $\{G_i^{**}\}$ can be obtained by projecting the G_i^* onto the set of Euclidean transformations (or any other desirable subset of $GL(d,\mathbb{R})$).

If $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ is close to being transitively consistent, the $d \times d$ block matrices in V_1 are invertible and can be identified with the G_1^{*-T}. This is guaranteed by the following lemma [23].

Lemma 11 *In this lemma Z or $Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})$ is fixed, whereas the matrix $\tilde{Z}$ is regarded as a variable in $\mathbb{R}^{nd\times nd}$. Let*

$$\mathcal{S}_1 = \{U \in \mathbb{R}^{nd\times d} : U^TU = I\},$$
$$\mathcal{S}_2(\tilde{Z}) = \arg\min_{U\in\mathcal{S}_1} \mathrm{trace}(U^T\tilde{Z}^T\tilde{Z}U).$$

For $\epsilon > 0$, there is $\delta(\epsilon) > 0$ such that if

$$\|\tilde{Z} - Z\|_{\mathrm{F}} < \delta,$$

it holds that for all $U \in \mathcal{S}_2(\tilde{Z})$,

$$\|U\|_{\mathcal{S}_2(Z)} < \epsilon,$$

where

$$\|U\|_{\mathcal{S}_2(Z)} = \inf_{V\in\mathcal{S}_2(Z)} \|U - V\|_F.$$

3.3 Further results

Loops in the graph $\mathcal{G}$ are essential for the performance of Algorithm 1 – if the graph is QSC and has no loops, improvement is not possible, see the following lemma.

Lemma 12 *If the QSC graph $\mathcal{G}$ is a spanning tree (containing a center), any collection $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ of matrices in $GL(d,\mathbb{R})$ is transitively consistent for $\mathcal{G}$.*

Proof:

$$Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}}) = \begin{bmatrix} I & * & * & \dots & * & * \\ 0 & I & * & \dots & * & * \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & I & * \\ 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix},$$

where all but one of the $*$ at each (block) row is nonzero and an invertible matrix. Due to this structure, there is a collection $\{G_i^*\}_{i\in\mathcal{V}}$ of matrices in $GL(d,\mathbb{R})$ such that (4) holds, which in turn means that $G_{ij} = G_i^{*-1}G_j$ for all $(i,j) \in \mathcal{E}$. Now since $\mathcal{G}$ is QSC, this means that $G_{ij} = G_i^{*-1}G_j^*$ for all $(i,j) \in \mathcal{V}\times\mathcal{V}$. ■

Due to Lemma 12, if $\mathcal{G}$ is QSC and a spanning tree and if $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ corresponds to "disturbed" versions of $\{G_{ij}^*\}_{(i,j)\in\mathcal{E}}$, the solution to Algorithm 1 will only provide the $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ once again.

Lemma 11 provides us with the positive result that the solution to Algorithm 1 depends continuously on the G_{ij} transformations. A somewhat negative result is provided by Lemma 13 below. Unfortunately it is not true that (3) implies transitive consistency.

Lemma 13 *Let $\mathcal{G} = (\mathcal{V},\mathcal{E})$ be any QSC graph satisfying that at least one element in the vector $A(\mathcal{G})[1,1,\dots,1]^T$ is greater or equal to 2. Let $\{G_{ij}^*\}_{(i,j)\in\mathcal{E}}$ be a collection of matrices in $GL(d,\mathbb{R})$, transitively consistent for $\mathcal{G}$. Let $\{G_i^*\}_{i\in\mathcal{V}}$ be a collection of matrices in $GL(d,\mathbb{R})$ for which it holds that*

$$G_{ij}^* = G_i^{*-1}G_j^* \text{ for all } (i,j)\in\mathcal{E}.$$

Now, for any $\epsilon > 0$, there is a collection $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ of matrices in $GL(d,\mathbb{R})$ that are not transitively consistent for $\mathcal{G}$ such that

$$\sum_{(i,j)\in\mathcal{E}} \|G_{ij} - G_{ij}^*\|_F \le \epsilon, \tag{5}$$

and (3) holds for $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ and a collection $\{G_i\}_{i\in\mathcal{V}}$ of matrices in $GL(d,\mathbb{R})$.

Proof:
Suppose the kth element of the vector $A(\mathcal{G})[1,1,\dots,1]^T$ is larger or equal to 2. Then there is l,m such that $l\neq k$, $m\neq k$, $G_{kl}^*, G_{km}^* \in GL(d,\mathbb{R})$. For $0<\alpha<1$ let $G_{kl} = (1+\alpha)G_{kl}^*$ and $G_{km} = (1-\alpha)G_{km}^*$. Furthermore, let $G_{ij} = G_{ij}^*$ for all $(i,j) \notin \{(k,l),(k,m)\}$. It is easy to see that the left-hand side of (5) is less than or equal to $\alpha(\|G_{kl}^*\|_F + \|G_{km}^*\|_F)$. Now we choose

$$\alpha < \frac{\epsilon}{\|G_{kl}^*\|_F + \|G_{km}^*\|_F}$$

and (5) is satisfied. By construction, all the G_{ij} are elements of $GL(d,\mathbb{R})$.

Let $G_i = G_i^*$ for all i. It holds that

$$\begin{aligned} & Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}}) \\ = \; & \mathrm{diag}(G_1^{-1}, G_2^{-1}, \dots, G_n^{-1})((L+Q)\otimes I)\cdot \\ & \mathrm{diag}(G_1, G_2, \dots, G_n), \end{aligned}$$

where $Q = [Q_{ij}]$, $Q_{kl} = \alpha$, $Q_{km} = -\alpha$ and $Q_{ij} = 0$ for all $(i,j) \notin \{(k,l),(k,m)\}$. Since $\ker((L+Q)\otimes I) \supset \ker(L\otimes I)$, (3) holds for the G_i. According to Lemma 8, if the G_{ij} are transitively consistent and $\mathcal{G}$ is QSC, (3) is a condition to guarantee (2). But (2) is not fulfilled since $G_kG_{kl}G_l^{-1} = (1+\alpha)I$. Thus, the G_{ij} are not transitively consistent. ■

After the introduction of Lemma 13, one might be lead to believe that Algorithm 1 does not work well in practice. However, as will be seen in Section 5, this is definitely not the case.

Now, to recap: Transitive consistency is equivalent to (2). Lemma 8 states that when $\mathcal{G}$ is QSC and transitive consistency holds, (2) and (3) are equivalent. However, Lemma 13 states that (3) is not equivalent to transitive consistency for QSC graphs.

Now we show a stability property of $-Z$. If the $\mathcal{G}_{ij}$ are transitively consistent, it is easy to see (from (4)) that $-Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})$ is critically stable, see definition in Lemma 14 below. However, the following result shows that if the G_{ij} transformations are elements in $O(d)$, i.e., $G_{ij}^TG_{ij} = I$ for all i,j, the matrix $-Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})$ is critically stable regardless if transitive consistency is fulfilled or not.

Lemma 14 *For any graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$ and collection $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ where $G_{ij} \in O(d)$ for all $(i,j)\in\mathcal{E}$, the matrix $-Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})$ is critically stable, i.e., for any $\epsilon > 0$, there is $\delta(\epsilon)$ such that for $x(0) = x_0 \in \mathbb{R}^{nd}$, $\|x(0)\| < \delta$ it holds that*

$$\|x(t)\| < \epsilon,$$

when

$$\dot{x}(t) = -Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})x(t).$$

Furthermore, if there are eigenvalues exactly on the imaginary axis, those eigenvalues are equal to zero.

Proof:
Let

$$\dot{x}(t) = -Zx(t), \quad x(t) \in \mathbb{R}^{nd}, \tag{6}$$

where $x(0)$ is the initial state. We can write $x(t)$ as $x(t) = [x_1^T(t), x_2^T(t), \ldots, x_n^T(t)]^T$, where $x_i(t) \in \mathbb{R}^d$ for all i. Define the function

$$V(x) = \max_i (x_i^T x_i).$$

If there is some eigenvalue of Z with negative real part or if there is a Jordan block of dimension larger than one corresponding to an eigenvalue on the imaginary axis, there is x_0 such that for the state $x(t)$ with initial state x_0, $V(x(t)) \to \infty$ as $t \to \infty$. We want to show that this is not possible. Let us first define the set

$$\mathcal{I}_{\max}(t) = \{i : V(x(t)) = x_i^T(t) x_i(t)\}.$$

Now,

$$\begin{aligned} D^+(V(x(t))) &= \max_{i \in \mathcal{I}_{\max}(t)} \frac{d}{dt} x_i^T(t) x_i(t) \qquad (7) \\ &= \max_{i \in \mathcal{I}_{\max}(t)} x_i^T(t) \left(\sum_{j \in \mathcal{N}_i} (G_{ij} x_j(t) - x_i(t) \right) \\ &\leq 0, \end{aligned}$$

where D^+ is the upper Dini-derivative. A proof of the first equality (7) can be found in [29] using the results in [30] and [31]. The result appears frequently in the literature [32,33]. Now we can use the Comparison Lemma [34] to show that $V(x(t))$ is decreasing independently of the choice of x_0. The inequality in (7) is a consequence of the fact that the G_{ij} are orthogonal matrices.

Now we show that there are no non-zero eigenvalues on the imaginary axis. Suppose there are non-zero eigenvalues on the imaginary axis, then there must be a nontrivial periodic solution $\bar{x}(t) = [\bar{x}_1^T, \bar{x}_1^T, \ldots, \bar{x}_n^T]^T$ to (6), i.e., $\bar{x}(t)$ is periodic and $\bar{x}(t_1) \neq \bar{x}(t_2)$ for some $t_1 \neq t_2$. It can be shown that $D^+(V(\bar{x}(t))) = 0$ for all t and it can also be shown that a necessary condition for this to hold is that $\bar{x}_i(t) = \bar{x}_i(t)$ for all t and $G_{ij} = I$ for all (i, j). The procedure to show the latter is a bit intricate and is based on an induction argument hinging on the fact that $\mathcal{G}$ is QSC. Now, if the $G_{ij} \neq I$, the necessary condition is not fulfilled, hence we have a contradiction. In the case when the $G_{ij} = I$ it holds that $Z(\mathcal{G}) = L(\mathcal{G}) \otimes I_d$ and the latter matrix does not have any non-zero eigenvalues on the imaginary axis. ■

3.4 Optimization problems and the H-matrix

In this subsection a matrix H is defined as the Hessian of a quadratic convex function. In the previous subsection the approach was to define a set of linear constraints, which are fulfilled for transitively consistent transformations, and then use these constraints to formulate a least squares optimization problem. In this section the approach is different. Optimization problems are formulated directly, without taking a detour via algebraic constraints. An assumption throughout this section is that $\mathcal{G}$ is connected.

Given the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and the collection $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ of matrices in $GL(d, \mathbb{R})$, we formulate three optimization problems, where the the first, (P1), corresponds to the exact problem we want to solve. The objective function is non-convex and the constraint set is non-compact. The second problem (P2) is a restriction of the first problem having a compact constraint set (with a non-convex objective function). In contrast, the third problem (P3) has a quadratic convex objective function of the G_i^{-1} as well as a compact constraint set.

$$\text{(P1)} \quad \begin{cases} \min\limits_{\{G_i\}_{i\in\mathcal{V}}} \sum\limits_{(i,j)\in\mathcal{E}} \frac{1}{2}\|G_{ij} - G_i^{-1}G_j\|_F^2, \\ \text{s.t.} \quad G_i \in GL(d, \mathbb{R}). \end{cases}$$

$$\text{(P2)} \quad \begin{cases} \min\limits_{\{G_i\}_{i\in\mathcal{V}}} \sum\limits_{(i,j)\in\mathcal{E}} \frac{1}{2}\|G_{ij} - G_i^{-1}G_j\|_F^2, \\ \text{s.t.} \quad G_i \in O(d). \end{cases}$$

$$\text{(P3)} \quad \begin{cases} \min\limits_{\{G_i\}_{i\in\mathcal{V}}} \sum\limits_{(i,j)\in\mathcal{E}} \frac{1}{2}\|G_{ij}G_j^{-1} - G_i^{-1}\|_F^2, \\ \text{s.t.} \quad U_1(\{G_i\}_{i\in\mathcal{V}})^T U_1(\{G_i\}_{i\in\mathcal{V}}) = Q \succ 0. \end{cases}$$

Define the two functions

$$\begin{aligned} f(\{G_i^{-1}\}_{i\in\mathcal{V}}) &= \sum_{(i,j)\in\mathcal{E}} \frac{1}{2}\|G_{ij}G_j^{-1} - G_i^{-1}\|_F^2, \\ g(\{G_i\}_{i\in\mathcal{V}}) &= \sum_{(i,j)\in\mathcal{E}} \frac{1}{2}\|G_{ij} - G_i^{-1}G_j\|_F^2. \end{aligned} \tag{8}$$

The matrix Q is symmetric and positive definite. We implicitly assume that g and f are parameterized by $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$. There is a similar problem to (P3), defined by left-multiplication by the G_i instead of right-multiplication by the G_i^{-1}:

$$\text{(P4)} \quad \begin{cases} \min\limits_{\{G_i\}_{i\in\mathcal{V}}} \sum\limits_{(i,j)\in\mathcal{E}} \frac{1}{2}\|G_iG_{ij} - G_j\|_F^2, \\ \text{s.t.} \quad U_2(\{G_i\}_{i\in\mathcal{V}})U_2(\{G_i\}_{i\in\mathcal{V}})^T = Q \succ 0, \end{cases}$$

The two problems are equivalent. We choose to study (P3) instead of (P4) in order to more easily see the connection between the Hessian (the H-matrix) in the problem and the matrix Z (cf. Section 3.5).

The problem (P2) and variants thereof has received attention lately [1]. Exact solutions do not exist in general and local gradient descent methods are used. One of the more important contributions of this work is that we provide a lower bound for the global solution of this problem as well as a method for which the bound is almost tight in numerical experiments.

3.5 Problem (P3) and its connection to problem (P1) – definition of the H-matrix

Let $X = U_1(\{G_i\}_{i\in\mathcal{V}})$. By a slight abuse of notation, let

$$f(X) = f(\{G_i^{-1}\}_{i\in\mathcal{V}}) = \sum_{(i,j)\in\mathcal{E}} \frac{1}{2}\|G_{ij}^T G_i^{-1} - G_j^{-1}\|_F^2.$$

Now

$$\nabla f(X) = X^T H(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}}),$$

where

$$\begin{aligned} &H(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}}) \\ &= Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}}) + Z_2(\bar{\mathcal{G}}, \{\bar{G}_{ij}\}_{(i,j)\in\bar{\mathcal{E}}}); \end{aligned}$$

and

$$\begin{aligned} &Z_2(\bar{\mathcal{G}}, \{\bar{G}_{ij}\}_{(i,j)\in\bar{\mathcal{E}}}) \\ =\ &\mathrm{diag}(W(\bar{\mathcal{G}}, \{\bar{G}_{ij}\}_{(i,j)\in\bar{\mathcal{E}}}) W(\bar{\mathcal{G}}, \{\bar{G}_{ij}\}_{(i,j)\in\bar{\mathcal{E}}})^T) \\ &- W(\bar{\mathcal{G}}, \{\bar{G}_{ij}\}_{(i,j)\in\bar{\mathcal{E}}}); \end{aligned}$$

$\bar{G}_{ij} = G_{ji}^T$ for all i, j and $\bar{\mathcal{G}} = (\mathcal{V}, \bar{\mathcal{E}})$ is the graph constructed reversing the direction of the edges in $\mathcal{G}$. The operator $\mathrm{diag}(\cdot)$ is here understood in the block-matrix sense, i.e., for a matrix $B \in \mathbb{R}^{nd\times nd}$, $\mathrm{diag}(B) = (I_n \otimes 1_d 1_d^T) \odot B$, where $\odot$ denotes element-wise multiplication, I_n is the n-dimensional identity matrix and 1_d is the d-dimensional vector containing ones.

Remark 15 *A more general formulation of the objective functions f and g with positive weights is*

$$\tilde{f}(\{G_i^{-1}\}_{i\in\mathcal{V}}) = \sum_{(i,j)\in\mathcal{E}} \frac{1}{2} a_{ij}\|G_{ij}G_j^{-1} - G_i^{-1}\|_F^2,$$

$$\tilde{g}(\{G_i\}_{i\in\mathcal{V}}) = \sum_{(i,j)\in\mathcal{E}} \frac{1}{2} a_{ij}\|G_{ij} - G_i^{-1}G_j\|_F^2.$$

The a_{ij} are positive for all i, j. This way of defining the objective functions lead to a slight modification of the H-matrix. Equivalent results to all the results obtained in this section for the H-matrix can also be formulated for this alternative definition of the H-matrix with weights. The alternative H-matrix can also be used in an equivalent distributed algorithm to the one presented in Section 4.1.2.

Lemma 16 *In the special case when all the G_{ij} are elements of $O(d)$, i.e., orthogonal matrices,*

$$Z_2(\bar{\mathcal{G}}, \{\bar{G}_{ij}\}_{(i,j)\in\bar{\mathcal{E}}}) = Z(\bar{\mathcal{G}}, \{\bar{G}_{ij}\}_{(i,j)\in\bar{\mathcal{E}}}).$$

Furthermore, if the graph $\mathcal{G}$ is also symmetric,

$$Z_2(\bar{\mathcal{G}}, \{\bar{G}_{ij}\}_{(i,j)\in\bar{\mathcal{E}}}) = Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})^T.$$

Lemma 17 *For any connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and collection $\{G_{ij}^*\}_{(i,j)\in\mathcal{E}}$ of matrices in $GL(d, \mathbb{R})$, the collection $\{G_{ij}^*\}_{(i,j)\in\mathcal{E}}$ is transitively consistent for $\mathcal{G}$ if and only if there is a collection $\{G_i^*\}_{i\in\mathcal{V}}$ of matrices in $GL(d, \mathbb{R})$ such that*

$$im(U_1(\{G_i^*\}_{i\in\mathcal{V}})) \subset ker(H(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}})).$$

The collection $\{G_i^*\}_{i\in\mathcal{V}}$ satisfies (2).

Proof: Suppose $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ is transitively consistent, then, according to Lemma 6, there is $\{G_i^*\}_{i\in\mathcal{V}}$ such that (2) holds, which in turn can be used to show that

$$\begin{aligned} &f(\{G_i^{*-1}\}_{i\in\mathcal{V}}) \\ =\ &U_1(\{G_i^*\}_{i\in\mathcal{V}})^T H(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}}) U_1(\{G_i\}_{i\in\mathcal{V}}) \\ =\ &0. \end{aligned} \tag{9}$$

On the other hand, if $\{G_{ij}^*\}_{(i,j)\in\mathcal{E}}$ is not transitively consistent, there is no $\{G_i^*\}_{i\in\mathcal{V}}$ such that (2) holds. It can now be shown that (9) does not hold for any collection $\{G_i^*\}_{i\in\mathcal{V}}$ of matrices in $GL(d, \mathbb{R})$. ■

Lemma 18 *For any connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and collection $\{G_{ij}^*\}_{(i,j)\in\mathcal{E}}$ of matrices in $GL(d, \mathbb{R})$ – transitively consistent for $\mathcal{G}$ – it holds that*

$$dim(ker(H(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}}))) = d.$$

Proof: Due to Lemma 17, we know that

$$\dim(\ker(H(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}}))) \geq d. \tag{10}$$

Thus, we need to show that the inequality in (10) cannot be strict. Since $\{G_{ij}^*\}_{(i,j)\in\mathcal{E}}$ is transitively consistent, there is $\{G_i^*\}_{i\in\mathcal{V}}$ fulfilling (2).

Suppose the inequality is strict for $\{G_{ij}^*\}_{(i,j)\in\mathcal{E}}$. We know there is $\{G_i^*\}_{i\in\mathcal{V}}$ where $G_i^* \in GL(d, \mathbb{R})$ for all i, such that

$$\mathrm{im}(U_1(\{G_i^*\}_{i\in\mathcal{V}})) \subset \ker(H(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}})).$$

Now there must be a vector $y = [y_1^T, y_2^T, \ldots, y_n^T]^T \in \mathbb{R}^{nd}$, where the y_i are in $\mathbb{R}^d$, such that

$$y \in \ker(H(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}})),$$

$y \neq 0$, and $y^T U_1(\{G_i^*\}_{i\in\mathcal{V}}) = 0$. There must be k and l such that the lth element of y_k is nonzero. The set of transformations $\{G_k^{*-1}G_i^*\}_{i\in\mathcal{V}}$ satisfy (2) (Lemma 6) and $f(\{(G_k^{*-1}G_i^*)^{-1}\}_{i\in\mathcal{V}}) = 0$. Now, let

$$\bar{X} = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_d] = U_1(\{G_k^{*-1}G_i^*\}_{i\in\mathcal{V}}),$$

and

$$\bar{Y} = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{l-1}, y, \bar{x}_{l+1}, \bar{x}_d] = U_1(\{G_k^{*-1}G_i^*\}_{i\in\mathcal{V}}).$$

We know that $H(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})\bar{Y} = 0$. For all i, let $\bar{Y}_i$ be the ith $d \times d$ block matrix in $\bar{Y}$. We know by construction that $\bar{Y}_k \in GL(d, \mathbb{R})$. Now, for any $j \in \mathcal{N}_k$ it holds that

$$\|G_{kj}^*\bar{Y}_j - \bar{Y}_k\|_F = 0,$$

which implies that $\bar{Y}_j \in GL(d, \mathbb{R})$. Also, for any i such that $k \in \mathcal{N}_i$, it holds that

$$\|G_{ik}^*\bar{Y}_k - \bar{Y}_i\|_F = 0,$$

which implies that $\bar{Y}_i \in GL(d, \mathbb{R})$. Now, due to the fact that $\mathcal{G}$ is connected, an induction argument can be used to show that all the $\bar{Y}_i$ are elements in $GL(d, \mathbb{R})$.

The collection $\{\bar{Y}_i\}_{i\in\mathcal{V}}$ satisfies

$$G_{ij}^* = \bar{Y}_i\bar{Y}_j^{-1} \text{ for all } (i,j) \in \mathcal{E}.$$

It is easy to see that the two collections $\{\bar{Y}_i^{-1}\}_{i\in\mathcal{V}}$ and $\{\bar{G}_i^*\}_{i\in\mathcal{V}}$ are not equal up to transformation from the left. But, since the graph is connected, the two must be equal up to transformation from the left (Lemma 6). This is a contradiction. Hence it is a false assumption that the inequality in (10) is strict. ■

We summarize the results of Lemma 17 and Lemma 18 in the following proposition.

Proposition 19 *The collection $\{G_{ij}^*\}_{(i,j)\in\mathcal{E}}$ is transitively consistent for the connected graph $\mathcal{G}$ if and only if there is a collection $\{G_i^*\}_{i\in\mathcal{V}}$ of matrices in $GL(d, \mathbb{R})$ such that*

$$im(U_1(\{G_i^*\}_{i\in\mathcal{V}})) = ker(H(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}})).$$

The G_i^* satisfy (2).

Proof: Direct application of Lemma 17 and Lemma 18. ■

The following proposition provides a similar, but somewhat stronger result.

Proposition 20 *The collection $\{G_{ij}^*\}_{(i,j)\in\mathcal{E}}$ of matrices in $GL(d, \mathbb{R})$ is transitively consistent for the connected graph $\mathcal{G}$ if and only if*

$$dim(ker(H(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}}))) = d.$$

Proof:
If: Let $\bar{Y} = [y_1, y_2, \ldots, y_{nd}]^T \in \mathbb{R}^{nd\times d}$ be any full rank matrix such that

$$H(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}})\bar{Y} = 0.$$

All the $y_i \in \mathbb{R}^d$. Let $\bar{Y}_i$ be the ith $d\times d$ block matrix in $\bar{Y}$. Since $\bar{Y}$ is full rank, there is a (finite) sequence $\{y_{i_j}\}_{j=1}^d$ such that $[y_{i_1}, y_{i_2}, \ldots y_{i_d}] \in GL(d, \mathbb{R})$.

Now, for $k \in \mathcal{V}$ we know that for any $j \in \mathcal{N}_k$ it holds that

$$\|G_{kj}^*\bar{Y}_j - \bar{Y}_k\|_F = 0,$$

which implies that $\text{im}(\bar{Y}_j^T) = \text{im}(\bar{Y}_k^T)$. Also, for any i such that $k \in \mathcal{N}_i$, it holds that

$$\|G_{ik}^*\bar{Y}_k - \bar{Y}_i\|_F = 0,$$

which implies that $\text{im}(\bar{Y}_i^T) = \text{im}(\bar{Y}_k^T)$. Now, due to the fact that $\mathcal{G}$ is connected, an induction argument can be used to show that $\text{im}(\bar{Y}_j^T) = \text{im}(\bar{Y}_i^T)$ for all i, j. But then

$$\text{im}([y_{i_1}, y_{i_2}, \ldots y_{i_d}]) \subset \text{im}(\bar{Y}_j^T) \text{ for all } j,$$

which together with the fact that $[y_{i_1}, y_{i_2}, \ldots y_{i_d}] \in GL(d, \mathbb{R})$ is full rank, can be used to show that $\bar{Y}_i \in GL(d, \mathbb{R})$ for all i. Thus,

$$\text{im}(U_1(\{\bar{Y}_i^{-1}\}_{i\in\mathcal{V}})) = \ker(H(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}})),$$

and the desired result follows from Proposition 19 where the G_i^* are replaced by the $\bar{Y}_i^{-1}$.

Only if: Direct application of Lemma 18. ■

Lemma 21 *The optimal solution to (P3) is*

$$X^* = VP,$$

where $P \in \mathbb{R}^{d\times d}$, and $V \in \mathbb{R}^{nd\times d}$. The matrix V is given by the solution to the problem

$$(P5) \quad \begin{cases} \min\limits_{W} trace(W^T H(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})W), \\ W \in \mathbb{R}^{nd\times d}, W^T W = I. \end{cases}$$

and P is given by the solution to the problem

$$\textit{(P6)} \quad \begin{cases} \min\limits_{\tilde{P}} \textit{trace}(\tilde{P}^T(V^T H(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})V)\tilde{P}), \\ \tilde{P} \in \mathbb{R}^{d\times d}, \tilde{P}^T\tilde{P} = Q. \end{cases}$$

Proof: In the new notation, problem (P3) is written as

$$\begin{cases} \min\limits_{W,\tilde{P}} \text{trace}(\tilde{P}^T(W^T H(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})W)\tilde{P}), \\ \tilde{P} \in \mathbb{R}^{d\times d}, \tilde{P}^T\tilde{P} = Q, W \in \mathbb{R}^{nd\times d}, W^TW = I. \end{cases}$$

In the following derivations it is assumed that $\tilde{P}$ and W belong to the constraint sets defined in the problem above.

$$\begin{aligned} & \min_{W,\tilde{P}} \text{trace}(\tilde{P}^T(W^T H(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})W)\tilde{P}) \\ = & \min_{W,\tilde{P}} \text{trace}(\tilde{P}^T(Q^T(W)D(W)Q(W))\tilde{P}) \\ = & \min_{W,\tilde{P}} \text{trace}(\tilde{P}^T D(W)\tilde{P}) \\ = & \min_{\tilde{P}} \text{trace}(\tilde{P}^T(V^T H(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})V)\tilde{P}), \end{aligned}$$

where $Q^T(W)D(W)Q(W)$ is the spectral factorization of

$$W^T H(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})W.$$

■

Proposition 22 *For any $Q_1 \succ 0$ and $Q_2 \succ 0$ let $\{G_i^*\}_{i\in\mathcal{V}}$ and $\{G_i^{**}\}_{i\in\mathcal{V}}$ be the transformations obtained from the optimal solutions of problem (P3) with Q equal to Q_1 and Q equal to Q_2 respectively. It holds that*

$$g(\{G_i^*\}_{i\in\mathcal{V}}) = g(\{G_i^{**}\}_{i\in\mathcal{V}}),$$

i.e., the value of g is independent of Q.

Proof: According to Lemma 21 the transformations are equal up to transformation from the left. ■

Remark 23 *It is implicitly assumed in Proposition 22 that the G_i^* and the G_i^{**} are in $GL(d,\mathbb{R})$. This is guaranteed if the G_{ij} are sufficiently close to be transitively consistent. The result to guarantee this is omitted but analogous to the statement in Lemma 11 for the Z-matrix.*

Lemma 24 *For any graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and collection $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ where $G_{ij} \in GL(d,\mathbb{R})$ for all $(i,j) \in \mathcal{E}$, the matrix $-H(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}})$ is critically stable, i.e., for any $\epsilon > 0$, there is $\delta(\epsilon)$ such that for any $x(0) = x_0 \in \mathbb{R}^{nd}$, $\|x(0)\| < \delta$ it holds that*

$$\|x(t)\| < \epsilon,$$

when

$$\dot{x}(t) = -H(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}})x(t).$$

Proof:
The matrix $H(\mathcal{G}, \{G_{ij}^*\}_{(i,j)\in\mathcal{E}})$ is the Hessian matrix and hence positive semi-definite. ■

3.6 A least squares method

Proposition 22 is important, it states that we can without loss of generality assume that $Q = I$, since the choice of Q does not affect the value of g, i.e., the cost function we want to minimize. The value of f changes with Q, but this is of less importance. Motivated by these results we introduce a least squares method along the lines of Algorithm 1.

Algorithm 2

(1) Let V_1 be the optimal solution to problem (P5).

(2) Identify the G_i^* in the collection $\{G_i^*\}_{i\in\mathcal{V}}$ by

$$V_1^T = [G_1^{*-T}, G_2^{*-T}, \ldots, G_n^{*-T}].$$

The algorithm is motivated by Proposition 19 and Proposition 22. If the collection $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ is close enough to be transitively consistent, step 2) can be executed, i.e., each $d \times d$ sub-block of the matrix V_1 is invertible. The result that guarantees this is analogous to the statement in Lemma 11.

3.7 A Gauss-Newton method

In this section a Gauss-Newton method is presented. The solution obtained in Algorithm 2 is used as the initialization for the algorithm.

The Fréchet derivatives of the identity map and the inverse map at the point G_i in the direction E_i are given by

$$\begin{aligned} L_{\text{id}}(G_i, E_i) &= E_i, \\ L_{\text{inv}}(G_i, E_i) &= -G_i^{-1}E_iG_i^{-1}, \end{aligned}$$

respectively. Higham [35] provides a good introduction to Fréchet derivatives for matrix functions. Let $\{E_i\}_{i\in\mathcal{V}}$ be a collection of matrices in $\mathbb{R}^{n\times n}$. It holds that

$$\begin{aligned} & g(\{G_i + E_i\}_{i\in\mathcal{V}}) \\ = & \sum_{(i,j)\in\mathcal{E}} \frac{1}{2}\|G_{ij} - G_i^{-1}G_j - G_i^{-1}L_{\text{id}}(G_j, E_j) \\ & - L_{\text{inv}}(G_i, E_i)G_j + o(\|[E_i, E_j]\|_F^2)\|_F^2. \end{aligned}$$

Let

$$\begin{aligned}&\bar{g}(\{G_i\}_{i\in\mathcal{V}},\{E_i\}_{i\in\mathcal{V}})\\ &=\sum_{(i,j)\in\mathcal{E}}\frac{1}{2}\|G_{ij}-G_i^{-1}G_j-G_i^{-1}L_{\mathrm{id}}(G_j,E_j)\\ &\quad-L_{\mathrm{inv}}(G_i,E_i)G_j\|_F^2\\ &=\sum_{(i,j)\in\mathcal{E}}\frac{1}{2}\|G_{ij}-G_i^{-1}G_j-G_i^{-1}E_j\\ &\quad+G_i^{-1}E_iG_i^{-1}G_j\|_F^2.\end{aligned}$$

Consider the following problem

$$\text{(P7)}\quad\Big\{\min_{\{E_i\}_{i\in\mathcal{V}}}\bar{g}(\{G_i\}_{i\in\mathcal{V}},\{E_i\}_{i\in\mathcal{V}}).$$

Problem (P7) is solved in each Gauss-Newton step of the method we present below (Algorithm 3). Its solution is given by the collection $\{E_i^*\}_{i\in\mathcal{V}}$ obtained by

$$\mathrm{vec}(U_2(\{E_i^*\}_{i\in\mathcal{V}}))=x,\tag{11}$$

where x is obtained by the solution of

$$\begin{aligned}&H_{\mathrm{GN}}(\{G_i\}_{i\in\mathcal{V}},\{G_{ij}\}_{(i,j)\in\mathcal{E}})x\\ =\;&-c_{\mathrm{GN}}(\{G_i\}_{i\in\mathcal{V}},,\{G_{ij}\}_{(i,j)\in\mathcal{E}});\end{aligned}\tag{12}$$

$\mathrm{vec}(\cdot)$ is the vectorization operator, i.e., it returns a vector with the stacked columns (in consecutive order) of its matrix-argument. The matrix $H_{\mathrm{GN}}\in\mathbb{R}^{nd^2\times nd^2}$ and the vector $c_{\mathrm{GN}}\in\mathbb{R}^{nd^2}$ are defined as follows (for simplicity we have omitted the explicit dependence of $\{G_i\}_{i\in\mathcal{V}}$ and $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$):

$$H_{\mathrm{GN}}=[\bar{H}_{ij}],$$

where $\bar{H}_{ij}\in\mathbb{R}^{d^2\times d^2}$ for all i,j. When $i\neq j$, $\bar{H}_{ij}$ is defined by

$$\bar{H}_{ij}=\begin{cases}0 & \text{if } \begin{cases}(j\notin\mathcal{N}_i),\\(i\notin\mathcal{N}_j),\end{cases}\\ -((G_i^{-1}G_j)\otimes(G_i^{-T}G_i^{-1})) & \text{if } \begin{cases}(j\in\mathcal{N}_i),\\(i\notin\mathcal{N}_j),\end{cases}\\ -((G_i^TG_j^{-T})\otimes(G_j^{-T}G_j^{-1})) & \text{if } \begin{cases}(j\notin\mathcal{N}_i),\\(i\in\mathcal{N}_j),\end{cases}\\ \begin{matrix}-((G_i^{-1}G_j)\otimes(G_i^{-T}G_i^{-1}))\\-((G_i^TG_j^{-T})\otimes(G_j^{-T}G_j^{-1}))\end{matrix} & \text{if } \begin{cases}(j\in\mathcal{N}_i),\\(i\in\mathcal{N}_j).\end{cases}\end{cases}$$

When $i=j$, $\bar{H}_{ii}$ is defined by

$$\begin{aligned}\bar{H}_{ii}=&\sum_{j\in\mathcal{N}_i}((G_i^{-1}G_jG_j^TG_i^{-T})\otimes(G_i^{-T}G_i^{-1}))+\\ &\sum_{\{j:i\in\mathcal{N}_j\}}I_d\otimes(G_j^{-T}G_j^{-1}).\end{aligned}$$

Now, $c_{\mathrm{GN}}=[c_1^T,c_2^T,\ldots,c_n^T]^T$, where $c_i\in\mathbb{R}^{d^2}$ for all i. The c_i are defined by

$$\begin{aligned}c_i=&\sum_{j\in\mathcal{N}_i}((G_i^{-1}G_j)\otimes G_i^{-T})\mathrm{vec}(G_{ij}-G_i^{-1}G_j)-\\ &\sum_{\{j:i\in\mathcal{N}_j\}}I_d\otimes G_j^{-T}\mathrm{vec}(G_{ji}-G_j^{-1}G_i).\end{aligned}$$

Algorithm 3

(1) Run Algorithm 2 and let $\{G_i^*\}_{i\in\mathcal{V}}$ bet the collection of matrices obtained in step (2) of that algorithm.

(2) Let $\mathbb{R}^{d\times d}\ni E_i^*=0$ for $i=1,2,\ldots,n$.

(3) **repeat:**

(a) $G_i\to G_i+E_i^*$ for all i,

(b) Update the E_i^* by (11) and (12), i.e.,

$$\mathrm{vec}(U_2(\{E_i^*\}_{i\in\mathcal{V}}))=x,$$

where x is the solution to (12).

The stoping criteria in step (3) of Algorithm 5 could be that the improvement of the cost function is smaller than a certain threshold for two consecutive iterations, or it could be that a certain number of iterations have been executed etc. It should be noted that H_{GN} is both positive definite and sparse. In order to solve (12) one can use for example the Conjugate Gradient method [36,37].

3.8 Affine and Euclidean transformations

In this subsection we consider affine and Euclidean transformations. These transformations are linear when homogenous coordinates are used. To be more precise, an element in $\mathrm{Aff}(d,\mathbb{R})$ is a matrix

$$G=\begin{bmatrix}Q & t\\ 0 & 1\end{bmatrix},$$

where $Q\in GL(d,\mathbb{R})$, $t\in\mathbb{R}^d$ and 1 is a scalar. Its inverse is given by

$$G^{-1}=\begin{bmatrix}Q^{-1} & -Q^{-1}t\\ 0 & 1\end{bmatrix}.$$

Euclidean transformations, $E(d)$, is a special case of affine transformations where the matrix $Q \in \mathcal{O}(d)$.

For any connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (due to Lemma 6), if and only if the collection $\{G_{ij}^*\}_{(i,j)\in\mathcal{E}}$ is transitively consistent and contains only affine transformations, there is a unique (up to transformation from the left by affine transformations) collection $\{G_i\}_{i\in\mathcal{V}}$ of affine transformations such that

$$G_{ij}^* = G_i^{-1} G_j.$$

Each G_i is given by

$$G = \begin{bmatrix} Q_i & t_i \\ 0 & 1 \end{bmatrix},$$

and each G_{ij}^* is given by

$$G_{ij}^* = \begin{bmatrix} Q_i^{-1}Q_j & Q_i^{-1}(t_j - t_i) \\ 0 & 1 \end{bmatrix}.$$

Now, let $\{G_{ij}^*\}_{(i,j)\in\mathcal{E}}$ be a collection of matrices in $\mathrm{Aff}(d, \mathbb{R})$ that are not necessarily transitively consistent. It holds that (by a slight abuse of notation)

$$\begin{aligned} g(\{G_i\}_{i\in\mathcal{V}}) &= \sum_{(i,j)\in\mathcal{E}} \frac{1}{2}\|G_{ij} - G_i^{-1}G_j\|_F^2 \\ &= g(\{Q_i\}_{i\in\mathcal{V}}) \\ &\quad + \sum_{(i,j)\in\mathcal{E}} \frac{1}{2}\|t_{ij} - Q_i^{-1}(t_j - t_i)\|_F^2, \quad (13) \end{aligned}$$

where t_{ij} is the translational part of the transformation G_{ij}. We see that there is a special structure of (13), where the cost function consists of two parts. The first part is only a function of the Q_i, whereas the second part is a function of both rotations and translations.

Define the following optimization problem

$$\text{(P8)} \quad \left\{ \min_{\{t_i\}_{i\in\mathcal{V}}} \sum_{(i,j)\in\mathcal{E}} \tfrac{1}{2}\|t_{ij} - Q_i^{-1}(t_j - t_i)\|_F^2. \right.$$

Let

$$c_{\mathrm{Aff}} = [c_1^T, c_2^T, \ldots, c_n^T]^T,$$

where

$$c_i = \sum_{j\in\mathcal{N}_i} Q_i^{-T} t_{ij} - \sum_{\{j:i\in\mathcal{N}_j\}} Q_j^{-T} t_{ji}.$$

Let

$$H_{\mathrm{Aff}} = [\tilde{H}_{ij}],$$

where $\tilde{H}_{ij} \in \mathbb{R}^{(d-1)\times(d-1)}$ for all i, j. When $i \neq j$, $\tilde{H}_{ij}$ is defined by

$$\tilde{H}_{ij} = \begin{cases} 0 & \text{if } \begin{cases} (j \notin \mathcal{N}_i), \\ (i \notin \mathcal{N}_j), \end{cases} \\ -Q_i^{-T}Q_i^{-1} & \text{if } \begin{cases} (j \in \mathcal{N}_i), \\ (i \notin \mathcal{N}_j), \end{cases} \\ -Q_j^{-T}Q_j^{-1} & \text{if } \begin{cases} (j \notin \mathcal{N}_i), \\ (i \in \mathcal{N}_j), \end{cases} \\ -Q_i^{-T}Q_i^{-1} - Q_j^{-T}Q_j^{-1} & \text{if } \begin{cases} (j \in \mathcal{N}_i), \\ (i \in \mathcal{N}_j). \end{cases} \end{cases}$$

When $i = j$, $\tilde{H}_{ii}$ is defined by

$$\tilde{H}_{ii} = \sum_{j\in\mathcal{N}_i} Q_i^{-T}Q_i^{-1} + \sum_{\{j:i\in\mathcal{N}_j\}} Q_j^{-T}Q_j^{-1}.$$

The matrix H_{Aff} and the vector c_{Aff} depend on $\{G_i\}_{i\in\mathcal{V}}$ and $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$.

The solutions to the problem (P8) is given by the elements in the set

$$\{\{t_i\}_{i\in\mathcal{V}} : H_{\mathrm{Aff}}[t_1^T, t_2^T, \ldots, t_n^T]^T = -c_{\mathrm{Aff}}\}.$$

The Gauss-Newton method developed in Section 3.7, i.e., Algorithm 3, can be adapted to the case of affine transformations. Now we require that

$$E_i \odot B = E_i \text{ for all } i, \text{ where } B = \begin{bmatrix} 1_d 1_d^T & 1_d \\ 0 & 0 \end{bmatrix}.$$

We remind the reader that $\odot$ denotes element-wise multiplication. In each iteration (in the modified step (3) of Algorithm 3) the collection $\{E_i^*\}_{i\in\mathcal{V}}$ is obtained by

$$\mathrm{vec}(U_2(\{E_i^*\}_{i\in\mathcal{V}})) = x, \quad (14)$$

where $x = Xv$, v is obtained by the solution to

$$X^T H_{\mathrm{GN}} X v = -X^T c_{\mathrm{GN}}, \quad (15)$$

and $X \in \mathbb{R}^{nd^2 \times n(d-1)d}$ is defined below.

$$X = I_n \otimes (I_d \otimes \bar{B}),$$

where

$$\bar{B} = \begin{bmatrix} I_{d-1} \\ 0 \end{bmatrix} \in \mathbb{R}^{d\times(d-1)}.$$

Now we present the following algorithm for affine transformations.

Algorithm 4

(1) Run Algorithm 2 for the collection $\{Q_{ij}\}_{(i,j)\in\mathcal{E}}$ and let $\{Q_i\}_{i\in\mathcal{V}}$ be the matrices obtained in step (2) of that algorithm.

(2) Solve problem (P8) for the t_i using the Q_i from (1) and let

$$G_i = \begin{bmatrix} Q_i & t_i \\ 0 & 1 \end{bmatrix} \text{ for all } i.$$

(3) Let $\mathbb{R}^{d\times d} \ni E_i^* = 0$ for $i = 1, 2, \ldots, n$.

(4) **while** a stoping criteria has not been met:

(a) $G_i \to G_i + E_i^*$ for all i,

(b) Update the E_i^* by (14) and (15), i.e.,

$$\text{vec}(U_2(\{E_i^*\}_{i\in\mathcal{V}})) = x,$$

where x is the solution to (14).

Remark 25 *There are many variations of Algorithm 4 that can be employed. The most simple one is to omit steps (3) and (4). Another one is to run the Gauss-Newton method (Algorithm 3) for the Q_i matrices after step (2). The expression in* (13) *can also be changed to include weights. For example, if the orthogonal matrices are closer to be transitively consistent than the translations, the first part of the expression, i.e., $g(\{Q_i\}_{i\in\mathcal{V}})$, could be weighted with a positive weight larger than* 1*.*

Remark 26 *After a slight modification, Algorithm 4 can be used for Euclidean transformations instead of affine ones. In this case Algorithm 5 (see Section 4) is used in (1) to generate the Q_i transformations instead of Algorithm 2. Numerical simulations (see Section 5) show that this is a good method in comparison to Algorithm 1 or Algorithm 2 (where the matrices are finally projected onto the set of Euclidean transformations $E(d)$).*

4 Orthogonal matrices

In this section problem (P2) is studied. For orthogonal matrices the objective functions f and g are equivalent. The Gauss-Newton method (Algorithm 3) is hence not necessary. Furthermore, the orthogonal matrices is an important class of matrices, not the least in dimension $d = 3$.

We begin by formulating the following result.

Proposition 27 *For the connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, let $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ be a collection of matrices in $GL(d, \mathbb{R})$. Let $\{G_i\}_{i\in\mathcal{V}}$ be a collection of matrices obtained from Algorithm 2. Let $\{G_i^*\}_{i\in\mathcal{V}}$ be a collection of matrices solving the optimization problem (P2). It holds that*

$$f(\{\sqrt{n}G_i^{-1}\}_{i\in\mathcal{V}}) \leq g(\{G_i^*\}_{i\in\mathcal{V}}).$$

Proof:
It is easy to verify that for orthogonal matrices, (P3) is a relaxation of (P2) when $Q = nI$. Now, (Proposition 22) the solution to (P3) with $Q = nI$ is provided by the matrices obtained by Algorithm 2 after scaling by $\frac{1}{\sqrt{n}}$. ■

Let us now extend Algorithm 1 (Algorithm 2) in the following way.

Algorithm 5

(1) Same as in Algorithm 1 (same as in Algorithm 2).

(2) Same as in Algorithm 1 (same as in Algorithm 2).

(3) Let G_i^{**} be the projection of G_j^* onto $O(d)$, i.e.,

$$G_i^{**} = QV^T,$$

where QDV^T is the SVD of G_i^*. Let

$$G_{ij}^* = G_i^{T**}G_j^{**}.$$

The collection $\{G_{ij}^*\}_{(i,j)\in\mathcal{E}}$ is the final transitively consistent collection.

Proposition 27 can now be used to provide performance guarantees. An upper bound on the closeness to optimality is given by

$$g(\{G_i^{**}\}_{i\in\mathcal{V}})) - f(\sqrt{n}G_i^{*-1}\}_{i\in\mathcal{V}}), \tag{16}$$

where the G_i^* are obtained from Algorithm 2 and the G_i^{**} are obtained from Algorithm 5 – assuming the first two steps are the same as in Algorithm 2.

If the G_{ij} are also elements in $O(d)$, the difference in (16) is almost tight. For example, in the case when $d = 3$, $n = 100$, and the G_{ij} are generated from G_i-matrices and R_{ij}-matrices matrices by $G_{ij} = G_i^{-1}G_jR_{ij}$ (R_{ij} is an orthogonal matrix with geodesic distance to I less or

equal to $\pi/4$. It is generated by drawing a skew symmetric matrix from the uniform distribution over the closed ball with radius $\pi/4$ and then taking the matrix exponential of that matrix). Let

$$\begin{aligned} &h(\{G_i^{**}\}_{i\in\mathcal{V}}, \{G_i^{*-1}\}_{i\in\mathcal{V}}) \\ =&\frac{g(\{G_i^{**}\}_{i\in\mathcal{V}})) - f(\{\sqrt{n}G_i^{*-1}\}_{i\in\mathcal{V}})}{f(\{\sqrt{n}G_i^{*-1}\}_{i\in\mathcal{V}})}. \end{aligned} \tag{17}$$

For 1000 experiments we observe that

$$h(\{G_i^{**}\}_{i\in\mathcal{V}}, \{G_i^{*-1}\}_{i\in\mathcal{V}}) \leq 6 * 10^{-4}.$$

This means that the solution obtained by Algorithm 5 is closer than 0.06% to the global optimum of problem (P2). The graphs in these experiments were QSC and the adjacency matrices contained 100 zero entries.

4.1 Distributed algorithms

In this subsection we show that Algorithm 5 can be implemented in a distributed way. Besides the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which describes what transformations are available, another graph $\mathcal{G}^{\text{com}} = (\mathcal{V}, \mathcal{E}^{\text{com}})$ is used. It is always assumed $\mathcal{E} \subset \mathcal{E}^{\text{com}}$. The graph $\mathcal{G}^{\text{com}}$ is referred to as *the communication graph*. The assumptions on the communication graph $\mathcal{G}^{\text{com}}$ differ between the two presented algorithms.

4.1.1 Orthogonal matrices and QSC communication graph

Here it is assumed that all transformations are orthogonal matrices, i.e., elements in $O(d)$. That is, the G_{ij} matrices as well as the G_{ij}^* matrices and the G_i^* matrices are assumed to be elements in $O(d)$.

The algorithm will now be presented, after which an explanation and justification is provided. In this algorithm it is assumed that $\mathcal{G} = \mathcal{G}^{\text{com}}$ is QSC. The notation $\mathcal{N}_i$ is used to denote $\mathcal{N}_i(\mathcal{G}) = \mathcal{N}_i(\mathcal{G}^{\text{com}})$.

Algorithm 6

Let

$$X(0) = [X_1^T(0), X_2^T(0), \ldots, X_n^T(0)]^T,$$

where, for all i, the elements of the matrix $X_i(0) \in \mathbb{R}^{d\times d}$ are drawn from $\mathcal{U}(-0.5, 0.5)$, i.e., the uniform distribution with the open interval $(-0.5, 0.5)$ as support. Let $X_i(t)$ for $t \in \mathbb{N}$ be defined by the following distributed algorithm:

$$\begin{aligned} X_1(t+1) &= X_1(t) + \epsilon \sum_{j\in\mathcal{N}_1} (G_{1j}X_j(t) - X_1(t)), \\ X_2(t+1) &= X_2(t) + \epsilon \sum_{j\in\mathcal{N}_2} (G_{2j}X_j(t) - X_2(t)), \\ &\vdots \\ X_n(t+1) &= X_n(t) + \epsilon \sum_{j\in\mathcal{N}_n} (G_{nj}X_j(t) - X_n(t)), \end{aligned}$$

where $\epsilon > 0.1$ In compact notation this is written as

$$X(t+1) = X(t) - \epsilon Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}}))X(t). \tag{18}$$

For a sufficiently large t, let G_i^{T*} be the projection of $X_i(t)$ onto $O(d)$, and let $G_{ij}^* = G_i^{T*}G_j^*$ for all i, j. It should be noted that if the spectral radius is not known, in practice it is enough to choose ϵ to something small.

Analysis of the algorithm

In this section the theoretical analysis of the algorithm is provided. The first thing we need to guarantee is that the matrix

$$I - \epsilon Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}}),$$

appearing in the right-hand side of the discrete time linear system (18), is critically stable in the linear dynamical systems sense. This means that all eigenvalues must be smaller than or equal to 1 in absolute value and any Jordan-block corresponding to an eigenvalue whose absolute value is 1 must be one-dimensional [38].

Lemma 28 *If $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is QSC and $\epsilon > 0$ small enough it holds that*

$$I - \epsilon Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})$$

is critically stable.

Proof: According to Lemma 14 it holds that Z is critically stable and has no non-zero eigenvalues on the imaginary axis. This means that for ϵ small enough the eigenvalues of $\epsilon Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}})$ are located in the closed unit disc centered at -1; the eigenvalues on the boundary are simple. ■

Remark 29 *Numerical simulations seem to indicate that in practice one can choose*

$$\epsilon \in \left(0, \frac{1}{\rho(Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}}))}\right),$$

where $\rho(Z(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}}))$ is the spectral radius.

Now we can deduce that if ϵ is chosen small enough, $X(t)$ converges (to something). It easy to verify (Lemma 8) that if the G_{ij} were transitively consistent, $X(t)$ would converge with exponential rate of convergence to

$$\bar{X} = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_d] = [\bar{X}_1^T, \bar{X}_2^T, \ldots, \bar{X}_n^T]^T,$$

where $\bar{x}_i$ is the projection of the ith column of $X(0)$ onto $\ker(Z)$ and $\bar{X}_i \in \mathbb{R}^{d\times d}$ for all i. Since the $X_i(0)$ are drawn from the distribution $(\mathcal{U}(-0.5, 0.5))^{d\times d}$, it is extremely unlikely that (probability zero) $\bar{X}$ has not full rank. If all the $\bar{X}_i$ are full rank matrices,

$$G_{ij} = \bar{X}_i \bar{X}_j^T \quad \text{for all } i, j.$$

Now, if the G_{ij} are not transitively consistent, in general the $X_i(t)$ converge to 0, which is not favorable. However, if the G_{ij} are close to being transitively consistent, since the eigenvalues of Z are continuous in the G_{ij}, the d smallest eigenvalues are significantly smaller in magnitude than the other eigenvalues; also the d smallest singular values are significantly smaller than the other singular values. Up to rotation, the right-singular vectors corresponding to the d smallest singular values are continuous in the G_{ij}, see Lemma 11.

Let the right-singular vectors corresponding to the d smallest singular values comprise the columns of the matrix $Y \in \mathbb{R}^{nd\times d}$. The matrix Y is equal to V obtained in the first step of Algorithm 1 (up to transformation from the left). Now, as $t \to \infty$, under the assumption that the G_{ij} are close to the G_{ij}^*, the columns of $X(t)$ converge to $\mathrm{im}(Y)$ much faster than $X(t)$ converges to 0. Thus, for t large enough $X(t)$ is approximately equal to Y up to transformation from the left. This convergence can be seen in Figure 9 for different choices of n, d, and magnitudes of noise.

The last step of the algorithm is justified by Lemma 8.

4.1.2 Orthogonal matrices and symmetric connected communication graph

In this section a general distributed algorithm is presented, which works for G_{ij} matrices in $GL(d, \mathbb{R})$, a directed connected graph $\mathcal{G}$, and a symmetric communication graph $\mathcal{G}^{\mathrm{com}}$. We will make the assumption that $\mathcal{G}^{\mathrm{com}}$ is the union graph of $\mathcal{G}$ and $\bar{\mathcal{G}}$, i.e., $\mathcal{G}^{\mathrm{com}} = (\mathcal{V}, \mathcal{E}\cup\bar{\mathcal{E}})$. The difference between Algorithm 6 and Algorithm 7 presented here, is that the Z-matrix is used in the former, whereas the H-matrix is used in the latter. Here, different from Section 4.1.1, it does not hold that $\mathcal{N}_i(\mathcal{G}^{\mathrm{com}}) = \mathcal{N}_i(\mathcal{G})$ for all i. When we write $\mathcal{N}_i$ this is shorthand for $\mathcal{N}_i(\mathcal{G}^{\mathrm{com}})$.

Algorithm 7

Let

$$X(0) = [X_1^T(0), X_2^T(0), \ldots, X_n^T(0)]^T,$$

where the elements of the matrix $X_i(0) \in \mathbb{R}^{d\times d}$ are drawn from $\mathcal{U}(-0.5, 0.5)$. Let $X_i(t)$ for $t \in \mathbb{N}$ be defined by the following distributed algorithm:

$$\begin{aligned}
X_1(t+1) &= X_1(t) + \epsilon \sum_{j\in\mathcal{N}_1} Q_{1j}X_j(t) - V_{1j}X_1(t)),\\
X_2(t+1) &= X_2(t) + \epsilon \sum_{j\in\mathcal{N}_2} (Q_{2j}X_j(t) - V_{2j}X_2(t)),\\
&\vdots\\
X_n(t+1) &= X_n(t) + \epsilon \sum_{j\in\mathcal{N}_n} (Q_{nj}X_j(t) - V_{nj}X_n(t)),
\end{aligned}$$

where

$$\epsilon \in \left(0, \frac{1}{\rho(H(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}}))}\right);$$

$\rho(H(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}}))$ is the spectral radius and

$$\begin{aligned}
Q_{ij} &= G_{ij} + G_{ji}^T,\\
V_{ij} &= I_d + G_{ji}^T G_{ji}.
\end{aligned}$$

In compact notation this is written as

$$X(t+1) = X(t) - \epsilon H(\mathcal{G}, \{G_{ij}\}_{(i,j)\in\mathcal{E}}))X(t). \quad (19)$$

For a sufficiently large t, let $G_i^{*-1} = X_i(t)$ and let $G_{ij}^* = G_i^{*-1}G_j^*$ for all i, j. It should be noted that if the spectral radius is not known, in practice it is enough to choose ϵ to something small.

Remark 30 *In the definitions of Q_{ij} and V_{ij}, in the case when $(j, i) \notin \mathcal{G}$, the symbol G_{ij} should be interpreted as the matrix in $\mathbb{R}^{d\times d}$ containing only zero-elements.*

Analysis of the algorithm

The convergence analysis of this Algorithm is analogous simpler than that of Algorithm 6. The eigenvalues of the H-matri are real since the matrix is symmetric. Instead of using Lemma 14, Lemma 24 is used instead.

4.2 Gradient flow for orthogonal matrices

Under the assumption that all the G_{ij} are elements in $O(d)$, we here provide a method, which will be used for comparison to our earlier methods. Results, along the lines of the ones presented in this section, can be found in [39,40,41].

For all i, define the cost functions

$$g_i : (O(d))^n \to \mathbb{R}^+$$

by

$$g_i(G_1, G_2, \ldots, G_n) = \sum_{j \in \mathcal{N}_i} A_{ij} \|G_i G_{ij} G_j^T - I\|_F^2.$$

The overall cost function

$$g : (O(d))^n \to \mathbb{R}^+$$

is equal to g, i.e.,

$$g(\{G_i^*\}_{i \in \mathcal{V}}) = \sum_{i=1}^{n} g_i(G_1, G_2, \ldots, G_n).$$

The (negative) gradient flow on $(O(d))^n$ of g is given by

$$\begin{aligned} \dot{G}_i(t) \ = & - \sum_{j \in \mathcal{N}_i} A_{ij}((G_i(t) G_{ij} G_j(t)^T)^T \\ & - (G_i(t) G_{ij} G_j(t)^T)) G_i(t) \\ & - \sum_{\{j : i \in \mathcal{N}_j\}} A_{ij}((G_j(t) G_{ij}^T G_i(t)^T)^T \\ & - (G_j(t) G_{ij}^T G_j(t)^T)) G_i(t), \\ & \text{for all } i \in \mathcal{V}. \end{aligned} \tag{20}$$

Now we present an algorithm, which improves on Algorithm 5. However, as will be seen in Section 5, this improvement is marginal.

Algorithm 8

(1) Run Algorithm 5 and let $\{G_i^{**}\}$ be the orthogonal matrices obtained in step (3) of the algorithm.

(2) Solve (20) numerically (for example by using ode45 in Matlab) for a sufficiently large time interval $[0, T]$ with the G_i^{**} as initial conditions.

(3) Let $\{G_i(t)^{-1} G_j(t)\}_{(i,j) \in \mathcal{V} \times \mathcal{V}}$ be the collection of transitively consistent matrices.

Remark 31 *In step (3) of Algorithm 8, if the $G_i(T)$ are not elements of $O(d)$ (due to errors from numerical integration), they need to be projected onto $O(d)$.*

5 Numerical verification

In our experiments, we consider Algorithm 8 first. Subsequently, the centralised Z- and H-matrix methods are evaluated for different configurations. Eventually, the analogous distributed methods are used in our simulations. In order to compare the methods, an assumption throughout this section is that the graph $\mathcal{G}$ – describing what transformations are available – is QSC.

5.1 Generating graphs and transformations

For each of the following experiments, the collection $\{G_i^*\}_{i \in \mathcal{V}}^n$ is generated by drawing random matrices in $O(d)$ [17]. From that, the (full) set of transitively consistent matrices $\{G_{ij}^* = G_i^{*-1} G_j^*\}_{i,j \in \mathcal{V}}$ is created. The noisy set of pairwise transformations $\{G_{ij}\}_{i,j \in \mathcal{V}}$ is generated by adding element-wise Gaussian noise with zero mean and standard deviation σ to each G_{ij}^*. After adding the element-wise Gaussian noise, the matrix is additionally projected onto $O(d)$.

Furthermore, a quasi-strongly connected (QSC) graph with graph density ρ – not mix up with the spectral radius of a matrix – is generated in the following manner. For generating a minimum QSC graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, two lists are used. One list $\mathcal{L}^{\mathcal{G}}$ keeps track of the nodes that are already considered, and one list $\mathcal{L}^{\bar{\mathcal{G}}}$ keeps track of the nodes that have not been considered. By a *minimum* QSC graph we mean a QSC graph that is a (spanning) tree, i.e., one with exactly $n-1$ edges. Initially, we set $\mathcal{V} = \{1, \ldots, n\}$, $\mathcal{E} = \{(i,i) : i \in \mathcal{V}\}$, $\mathcal{L}^{\mathcal{G}} = \{r\}$, where $r \in \mathcal{V}$ is a randomly selected node, and $\mathcal{L}^{\bar{\mathcal{G}}} = \{1, \ldots, n\} - \{r\}$. Then, the following procedure is repeated $n-1$ times: pick random nodes $i \in \mathcal{L}^{\bar{\mathcal{G}}}$ and $j \in \mathcal{L}^{\mathcal{G}}$, add the edge (i,j) to $\mathcal{E}$, and update $\mathcal{L}^{\bar{\mathcal{G}}}$ and $\mathcal{L}^{\mathcal{G}}$ accordingly. After $n-1$ repetitions a (minimum) QSC graph has been generated. At this point we store the edge set $\mathcal{E}$ and call it $\mathcal{E}^{QSC}$. Next, random edges are added to $\mathcal{E}$ until the the density of the graph is larger than or equal to ρ, which is defined below.

We remind the reader that A is the adjacency matrix of $\mathcal{G}$ with elements A_{ij}. The graph density $\rho(\mathcal{G})$ is defined by

$$\rho(\mathcal{G}) = \left(\frac{1}{n^2 - |\mathcal{E}^{QSC}|} \sum_{(i,j) \notin \mathcal{E}^{QSC}}^{n} A_{ij} \right) . \tag{21}$$

The intuition behind the graph density is that it is the proportion of the number of present edges in $\mathcal{G}$ with respect to a fully connected graph (having n^2 edges) excluding the edges in $\mathcal{E}^{QSC}$. With that, $\rho = 0$ denotes a minimum QSC graph, whereas $\rho = 1$ denotes a fully connected graph. Generating random QSC graphs with different values of the parameter ρ allows us to consider different degrees of missing transformations.

Using the graph $\mathcal{G}$ with density ρ, the collection $\{G_{ij}\}_{(i,j) \in \mathcal{E}}$ is the one that is eventually used for the

evaluation. In the simulations, for each individual sub-figure the simulations have been performed with 100 random sets of orthogonal transformations (the transitively consistent ones and the synthetically generated noisy versions thereof) and QSC graphs. Shown in the sub-figures is the mean of all runs.

5.2 Algorithm 8 – orthogonal matrices

For $d = 3$, Figure 1 shows upper bounds on the gap between the optimal value and the value of the objective function obtained by two methods – Algorithm 5, green curve, and Algorithm 8, blue curve. In Algorithm 8, the initial states are given by the solution to Algorithm 5. The ODE in (20) is solved numerically in Matlab by ode45. For each number of coordinate systems n, 100 simulations are conducted and averages are shown in Figure 1. In each simulation a set of transitively consistent orthogonal matrices $\{G_{ij}^*\}_{(i,j)\in\mathcal{V}\times\mathcal{V}}$ are generated from a set of orthogonal matrices $\{G_i^*\}_{i\in\mathcal{V}}$ according to the description in Section 5.1 below. The graph G used in each of the experiments is the complete graph.

For a single numerical experiment, Figure 2 shows the improvement of h when Algorithm 8 is used. One can see that Algorithm 5 generates matrices that are close to a local optimum of problem (P2).

It can be seen that only a marginal improvement can be made using the significantly more computationally expensive Algorithm 8. Due to the heavy computational burden, in the following simulations we omit Algorithm 8 and focus on the methods based on the Z- and H-matrix.

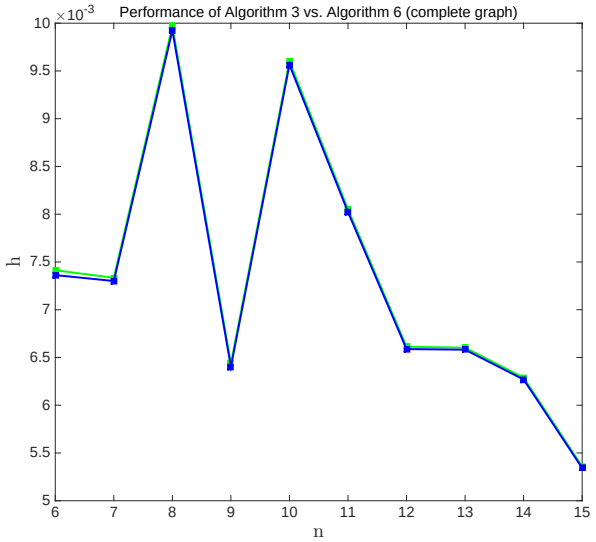

Fig. 1. Upper bounds on the optimality gap, i.e., h, for the solution of Algorithm 5, green line, and Algorithm 8, blue line. The graph is complete.

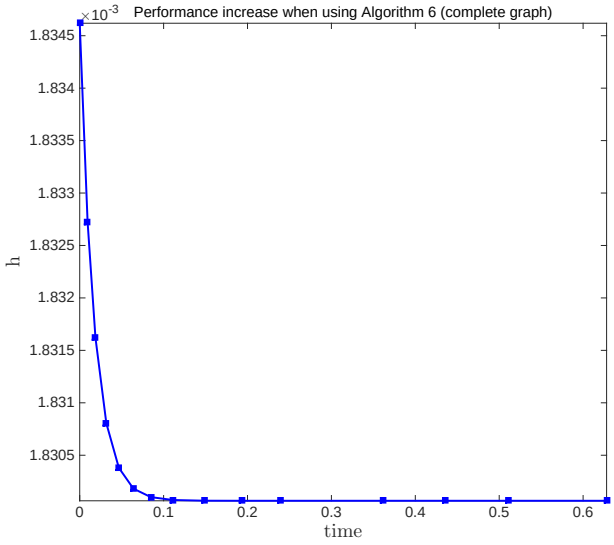

Fig. 2. Improvement of h when Algorithm 8 is used. In this case when $n = 15$, $d = 3$, and the graph is complete.

5.3 Centralized methods for matrices in $O(d)$

In this set of experiments we compare the H-matrix method, the Z-matrix method and a (naive) reference-based method, where the latter serves as baseline for the comparison.

5.3.1 The reference-based method

For the reference-based method, a minimum QSC graph $\mathcal{G}^{\text{min-QSC}} = (\mathcal{V}, \mathcal{E}^{\text{min-QSC}} \subseteq \mathcal{E})$ with $n-1$ edges is randomly drawn as a subgraph of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For that, all centers (see Def. 2) of the graph $\mathcal{G}$ are initially determined by looking at the $n \times n$ distance matrix between all n nodes. From the set of centers, a node c is randomly selected. Since $\mathcal{G}$ is QSC, there is at least one such center. Let $\mathcal{L}^{\mathcal{G}^{\text{min-QSC}}} = \{c\}$ be the list of nodes that have already been considered and initialise $\mathcal{E}^{\text{min-QSC}} = \emptyset$. The following procedure is repeated until $|\mathcal{E}^{\text{min-QSC}}| = n-1$: randomly select a node $r \in \mathcal{L}^{\mathcal{G}^{\text{min-QSC}}}$, select a random node $r' \in \{i : (i,r) \in \mathcal{E}\}$, if there is such an r', add the edge (r', r) to $\mathcal{E}^{\text{min-QSC}}$ and add r' to $\mathcal{L}^{\mathcal{G}^{\text{min-QSC}}}$.

Per construction, the graph $\mathcal{G}^{\text{min-QSC}}$ is a spanning tree that contains a center. Thus, according to Lemma 12, the set $\{G_{ij}\}_{(i,j)\in\mathcal{E}^{\text{min-QSC}}}$ is transitively consistent for $\mathcal{G}^{\text{min-QSC}}$. W.l.o.g., by setting $G_c^* = I$ for the center c of $\mathcal{G}^{\text{min-QSC}}$, all (other) G_i^* are (uniquely) determined as

$$G_i^* = G_j^* G_{ij}^{-1} \quad \text{for } i \neq c,\ (i,j) \in \mathcal{E}^{\text{min-QSC}}. \tag{22}$$

To summarise, in the reference-based method a (random) rooted spanning tree graph is considered as sub-

graph of $\mathcal{G}$, i.e., all but $n-1$ relative transformations G_{ij} (accounting for the transitive inconsistency) are discarded such that the remaining $n-1$ relative transformations are transitively consistent.

In Fig. 3 the results of the experiments are shown. On the vertical axis, a normalised version of the function in (8), defined by

$$g'(\{G_i\}_{i\in\mathcal{V}}) = \frac{1}{|\mathcal{E}|}\sum_{(i.j)\in\mathcal{E}} \|G_{ij} - G_i^{-1}G_j\|_F^2 \qquad (23)$$

is used. Each sub-figure shows a different varying parameter on the horizontal axis. The title of each sub-figure indicates the fixed parameters.

It can be seen that in all cases the Z-matrix approach is nearly as good as the H-matrix approach when looking at orthogonal transformations. However, as anticipated, the reference-based method performs worse than both proposed methods. For the case of different degrees of noise (Fig. 3, top left) it can be seen that the total error increases with increasing noise. Similarly, in the case of different dimensions (Fig. 3, bottom left), the error increases with increasing dimensionality. This can be explained by the fact that the Frobenius norm in (23) sums over d^2 values. For various values of the graph density (Fig. 3, top right), the error for the H- and Z-matrix method is approximately constant (apart from the case of a rooted spanning tree at $\rho = 0$, according to Lemma 12.).

5.4 Centralized methods for matrices in $GL(d,\mathbb{R})$

In this set of experiments we compare the H-matrix method and the Z-matrix method.

Using the reference-based method for the case of linear transformations is problematic because this method inverts the matrices G_{ij} for $(i,j) \in \mathcal{E}^{\text{min-QSC}}$ (see (23)). Therefore, for reasonably large noise, it is likely that there is some $(i,j) \in \mathcal{E}^{\text{min-QSC}}$ where G_{ij} is ill-conditioned, resulting in the corresponding term in g' blowing up. In Fig. 5 this problem is illustrated, where the horizontal axis is shown in log-scale. The lines of the Z- and H-matrix methods almost coincide, so only the green line of the H-matrix method is visible. The reference-based method's (black) line results in extremely large errors. Due to this reason, and since we have already shown that for the case of orthogonal transformations the reference-based method is inferior, in the following the reference based method is not used in the comparisons.

For the complete graph case, in Figure 7 the improved performance the Gauss-Newton method, i.e., Algorithm

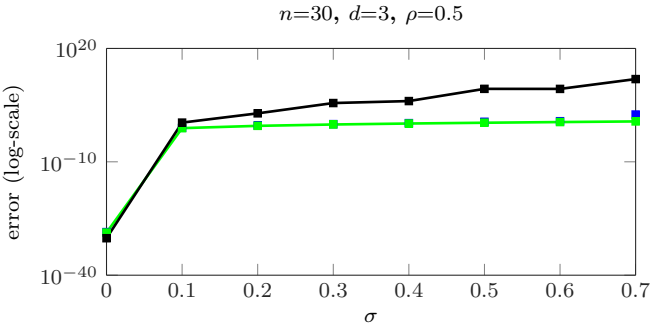

Fig. 5. Normalised error in (23) on the horizontal axis shown as log-scale for the Z-matrix method (blue), the H-matrix method (green) and the reference-based method (black) when considering transformations in $GL(d,\mathbb{R})$. Note that the blue and green line coincide.

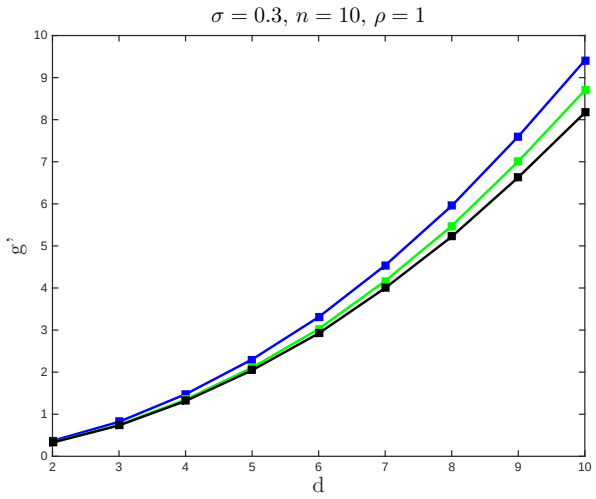

Fig. 7. Performance of the Z-matrix method (blue), the H-matrix method (green), and the Gauss-Newton method with the solution of the H-matrix method as initialization (black).

3, (with the solution of the H-matrix method as initialization) is shown. The Gauss-Newton method run 5 iterations, but from inspection it could be deduced that the main convergence occurs already after two iterations.

In Fig. 6, the comparisons of the H-matrix method and the Z-matrix method are shown. It can be seen that for small noise (Fig. 6, top left) both methods are comparable, whereas for a larger amount of noise the H-matrix method is able to obtain a smaller error. Similarly, for transformations with small dimensionality (Fig. 6, bottom left), both methods are comparable whereas for larger dimensions the gap between both approaches increases. On the contrary, (Fig. 6, top right) illustrates that with increasing graph density the line of the Z-matrix method approaches that of the H-matrix method (apart from the spanning tree case when $\rho = 0$, analogous to the orthogonal transformation experiments). This indicates that the H-matrix method performs bet-

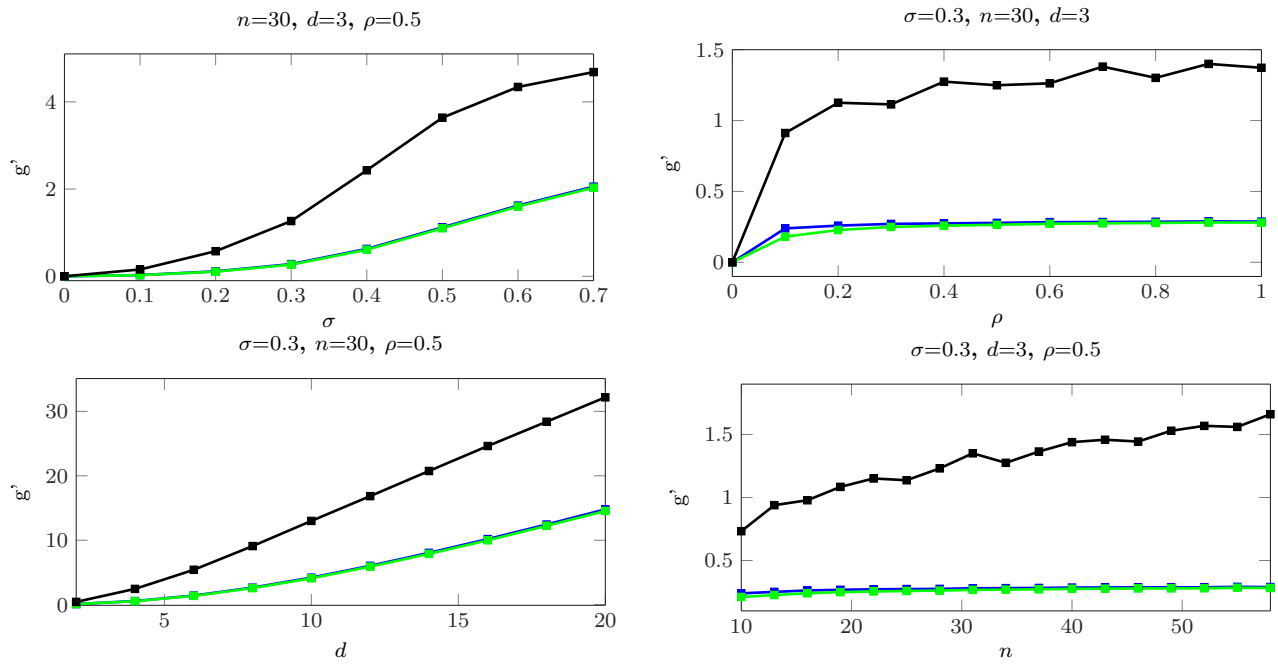

Fig. 3. Normalised error in (23) for the reference-based method (black), the Z-matrix method (blue) and the H-matrix method (green) when considering transformations in $O(d)$. In each sub-figure, a different parameter varies along the horizontal axis.

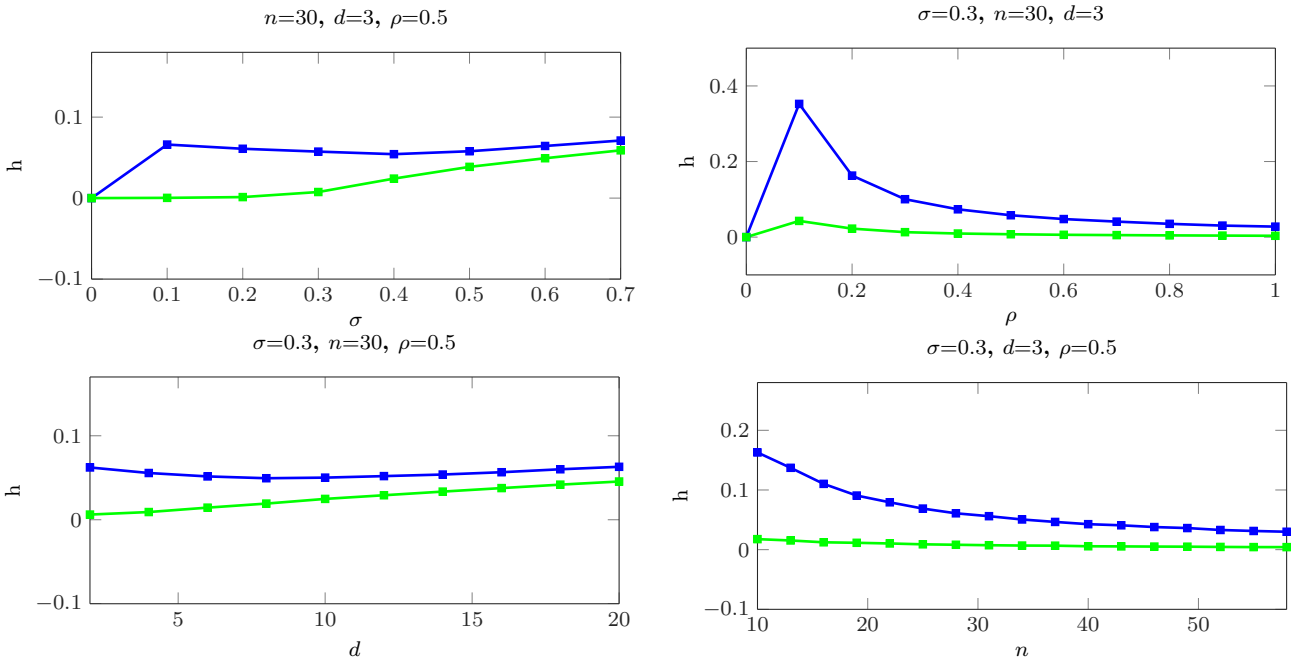

Fig. 4. Gap function in (17) for the Z-matrix method (blue) and the H-matrix method (green) when considering transformations in $O(d)$. In each sub-figure, a different parameter varies along the horizontal axis.

ter than the Z-matrix method if there is only little information available. A similar observation can be made for various n (Fig. 6, bottom right). For each subfigure, 100 simulations for a certain configuration of σ, n, d, and ρ are shown.

5.5 *Methods for affine and Euclidean transformations*

In Figure 8 – for affine and Euclidean transformations – a comparison between four different methods can be found. The G_{ij} transformations are affine respective Euclidean, but only the Algorithm 4 methods (red and

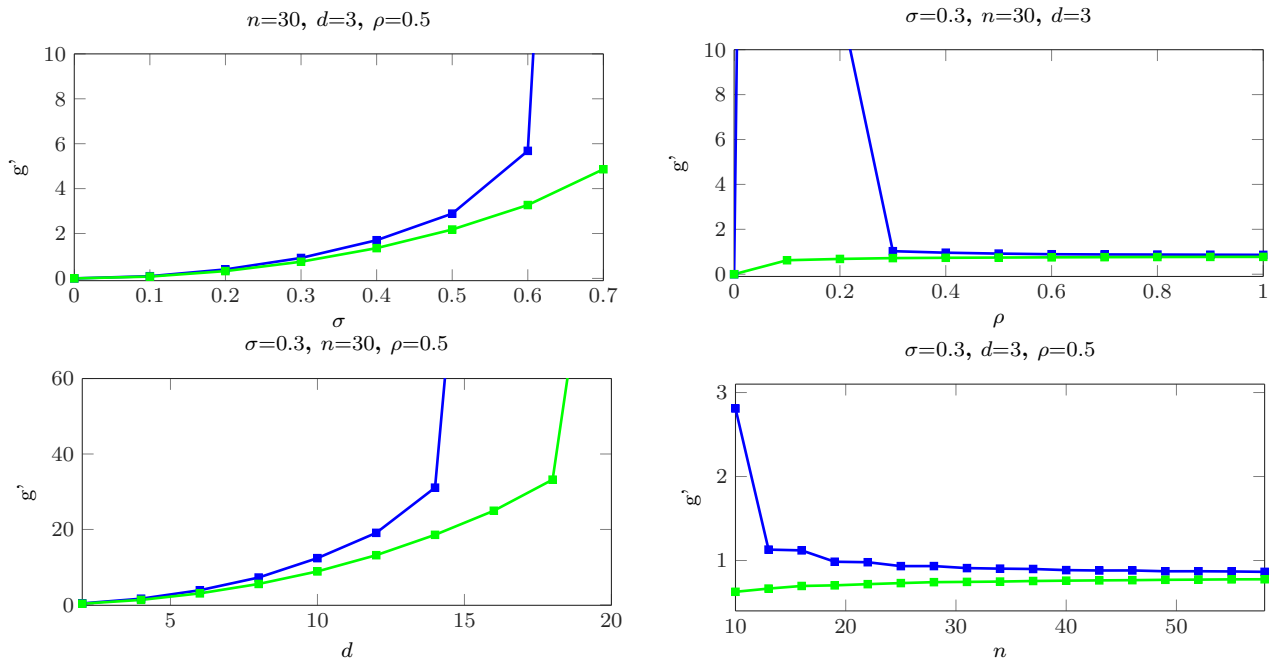

Fig. 6. Normalised error in (23) for the Z-matrix method (blue) and the H-matrix method (green) when considering transformations in $GL(d, \mathbb{R})$. In each sub-figure, a different parameter varies along the horizontal axis.

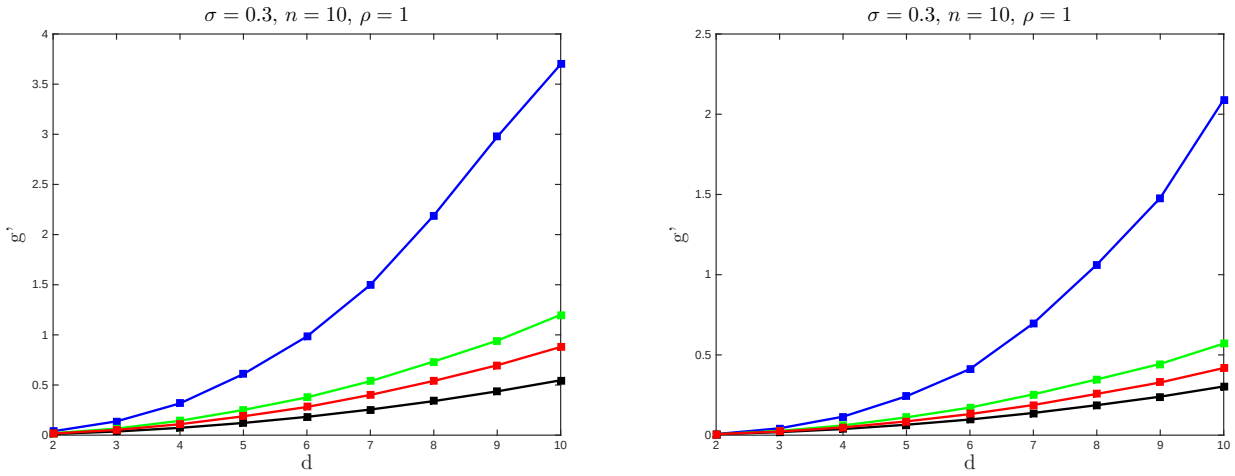

Fig. 8. Left figure: Performance of the methods for affine transformations. The Z-matrix method (blue), the H-matrix method (green), the first two steps of Algorithm 4 (red), and Algorithm 4 (black). Right figure: Performance of the methods for Euclidean transformations. The Z-matrix method (blue), the H-matrix method (green), the first two steps of Algorithm 4 where Algorithm 5 has been used instead of Algorithm 2 (red), and Algorithm 4 where Algorithm 5 has been used instead of Algorithm 2 (black).

black) preserve this property. In the bottom right figure the G_i transformations obtained in the Z-matrix method respective the H-matrix method have been projected onto $E(d)$, i.e., the set of Euclidean transformations. The orthogonal matrix part of the G_{ij} transformations were generated according to the description above. The elements in the transnational vectors were drawn from the uniform distribution over $(-5, 5)$ and additional element-wise noise was added.

5.6 Distributed methods

Results of the distributed Z-matrix method are shown in Fig. 9 and results for the distributed H-matrix method are shown in Fig. 10.

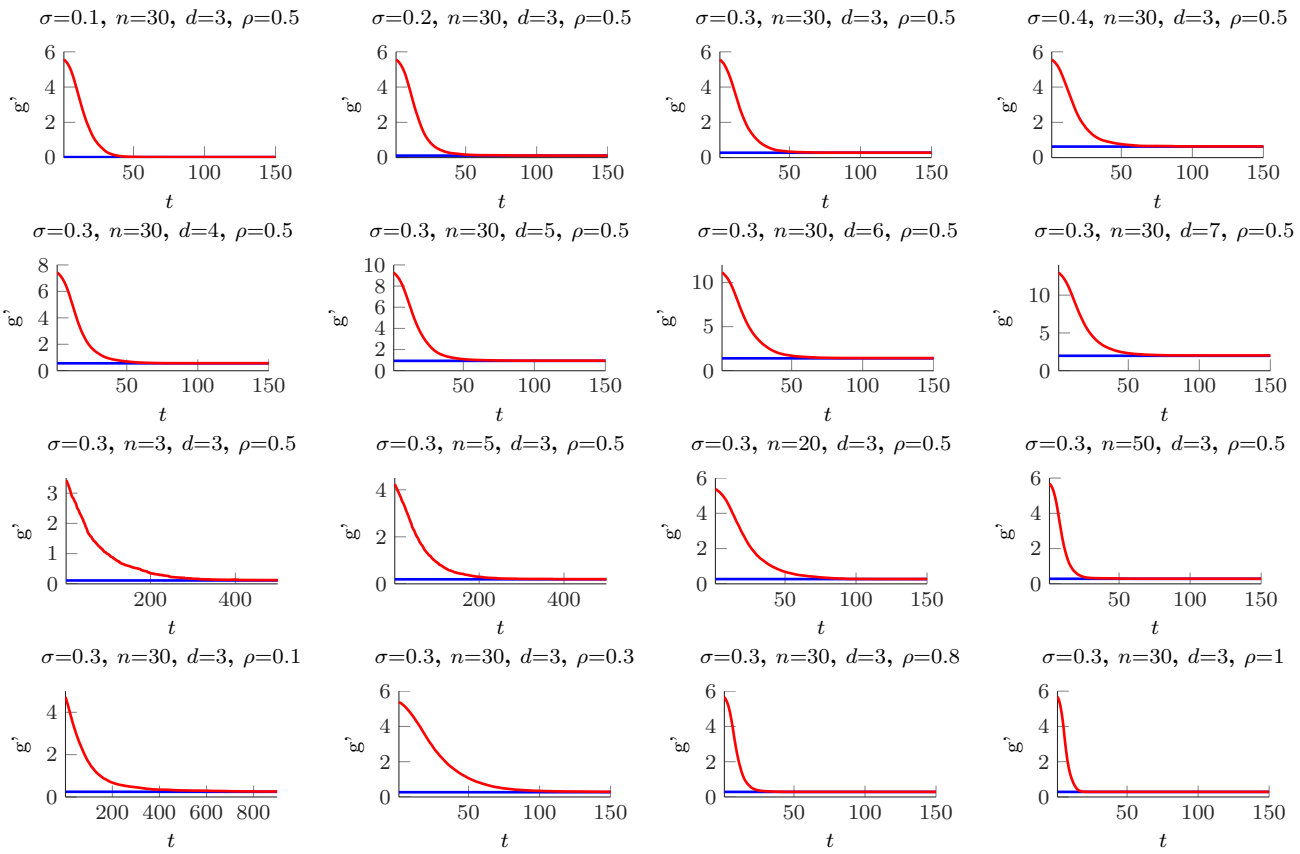

Fig. 9. Normalised error in (23) on the vertical axis for the Z-matrix method (blue) and its distributed version (red) when considering transformations in $O(d)$. The horizontal axis shows the number of steps, where the step size has been chosen as $\epsilon = 0.01$ in each sub-figure.

Conclusions

This worked addressed transitive consistency of linear inverible transformations between Euclidean coordinate systems. Given a set of linear invertible transformations (or matrices) – that are not transitively consistent – the proposed methods synchronize the transformations. This means that they provide transformations that are both transitively consistent and close to the original non-synchronized transformations. First two different direct or centralized approaches were proposed. In the first approach – the Z-matrix approach – linear algebraic conditions were formulated that must hold for transitively consistent transformations. Then the sought transformations are obtained from the solution of a least squares problem. In the second approach – the H-matrix approach – optimization problems were formulated directly, without taking a detour via linear algebraic constraints. The sought transformations are obtained from the solution of the optimization problems.

A Gauss-Newton iterative method was also proposed where the solution from the H-matrix method was used as initialization. This method was later adapted to the case of affine and Euclidean transformations. It was shown in numerical simulations that for the case of affine and Euclidean transformations, this approach outperforms the H-matrix approach and the Z-matrix approach. However, for orthogonal transformations no improvement is possible over the H-matrix method.

In a later part of the paper, for orthogonal matrices, two distributed algorithms were presented. These algorithms share similarities with linear consensus algorithms for distributed averaging. It was shown that these simple consensus-like protocols can be used to provide a solution to our problem that is very close to the global optimum – even for noise large in magnitude. The proposed methods – both the direct/centralized and the iterative/distributed – have been verified to work in numerical experiments for a wide range of parameter settings.

References

[1] R. Tron and R. Vidal. Distributed 3-d localization of camera sensor networks from 2-d image measurements. *Transactions on Automatic Control*, 59(12):3325–3340, 2014.

[2] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009.

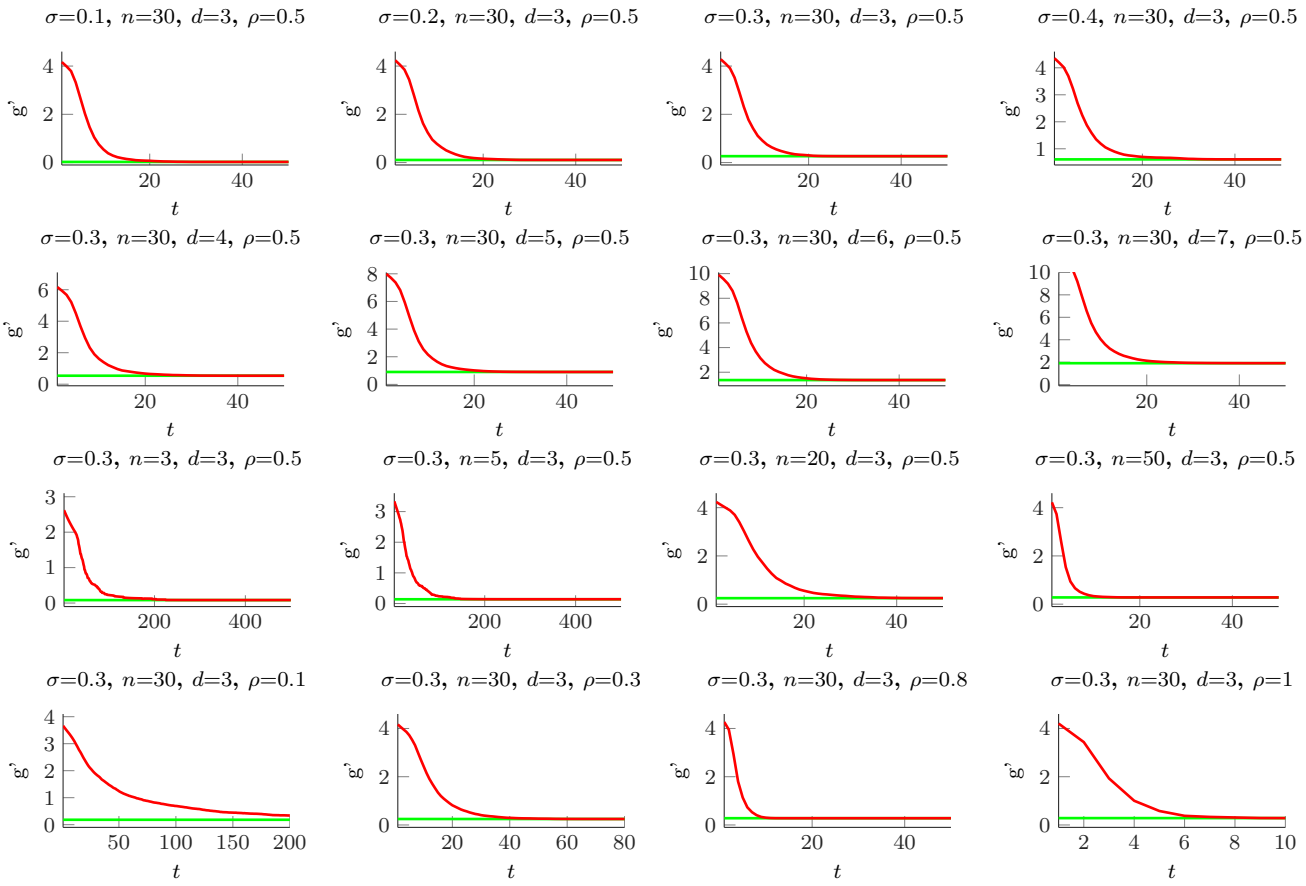

Fig. 10. Normalised error in (23) on the vertical axis for the H-matrix method (green) and its distributed version (red) when considering transformations in $O(d)$. The horizontal axis shows the number of steps, where the step size has been chosen as $\epsilon = 0.01$ in each sub-figure.

[3] O. Škrinjar, A. Bistoquet, and H. Tagare. Symmetric and transitive registration of image sequences. *Journal of Biomedical Imaging*, 2008:14, 2008.

[4] S. Joshi, B. Davis, M. Jomier, and G. Gerig. Unbiased diffeomorphic atlas construction for computational anatomy. *NeuroImage*, 23:S151–S160, 2004.

[5] T.F. Cootes, S. Marsland, C.J. Twining, K. Smith, and C.J. Taylor. Groupwise diffeomorphic non-rigid registration for automatic model building. In *Computer Vision-ECCV 2004*, pages 316–327. Springer, 2004.

[6] S. Allassonnière, Y. Amit, and A. Trouvé. Towards a coherent statistical framework for dense deformable template estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(1):3–29, 2007.

[7] F. Bernard, J. Thunberg, A. Husch, L. Salamanca, P. Gemmar, F. Hertel, and J. Goncalves. Transitively consistent and unbiased multi-image registration using numerically stable transformation synchronisation. In *Workshop on Spectral Analysis in Medical Imaging (SAMI), to appear*. MICCAI, 2015.

[8] T. Gass, G. Szekely, and O. Goksel. Detection and correction of inconsistency-based errors in non-rigid registration. In *SPIE Medical Imaging*, pages 90341B–90341B. International Society for Optics and Photonics, 2014.

[9] X. Geng. *Transitive inverse-consistent image registration and evaluation*. ProQuest, 2007.

[10] J.C. Gower and G.B. Dijksterhuis. *Procrustes problems*, volume 3. Oxford University Press Oxford, 2004.

[11] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):698–700, 1987.

[12] P.H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, March 1966.

[13] B.K.P. Horn, H.M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127, 1988.

[14] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

[15] M.W. Walker, L. Shao, and R.A. Volz. Estimating 3-D location parameters using dual number quaternions. *CVGIP: Image Understanding*, 54(3):358–367, November 1991.

[16] D. Pizarro and A. Bartoli. Global optimization for optimal generalized procrustes analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2409–2415. IEEE, 2011.

[17] F. Bernard, J. Thunberg, P. Gemmar, F. Hertel, A. Husch, and J. Goncalves. A solution for multi-alignment by transformation synchronisation. In *Conference on Computer Vision and Pattern Recognition*. IEEE, 2015.

[18] A. Singer and Y. Shkolnisky. Three-Dimensional Structure Determination from Common Lines in Cryo-EM by Eigenvectors and Semidefinite Programming. *SIAM journal on imaging sciences*, 4(2):543–572, June 2011.

[19] R. Hadani and A. Singer. Representation Theoretic Patterns in Three-Dimensional Cryo-Electron Microscopy II—The Class Averaging Problem. *Foundations of computational mathematics (New York, N.Y.)*, 11(5):589–616, 2011.

[20] R. Hadani and A. Singer. Representation theoretic patterns in three dimensional Cryo-Electron Microscopy I: The intrinsic reconstitution algorithm. *Annals of mathematics*, 174(2):1219, 2011.

[21] K.N. Chaudhury, Y. Khoo, and A. Singer. Global registration of multiple point clouds using semidefinite programming. *arXiv.org*, June 2013.

[22] D. Pachauri, R. Kondor, and V. Singh. Solving the multi-way matching problem by permutation synchronization. In *Advances in neural information processing systems*, pages 1860–1868, 2013.

[23] J. Thunberg, F. Bernard, and J. Goncalves. Centralized and distributed transformation synchronization under partial information. *Submitted to: Transactions on Pattern Analysis and Machine Intelligence*.

[24] M. Mesbahi and M. Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[25] A. Jadbabaie and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[26] R. Olfati-Saber and R. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

[27] R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[28] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

[29] J. Thunberg. *Consensus and Pursuit-Evasion in Nonlinear Multi-Agent Systems*. PhD thesis, KTH Royal Institute of Technology, 2014.

[30] T. Yoshizawa. *Stability theory by Liapunov's second method*. Mathematical Society of Japan (Tokyo), 1966.

[31] F.H. Clarke. Generalized gradients and applications. *Transactions of the American Mathematical Society*, 205:247–262, 1975.

[32] G. Shi and Y. Hong. Global target aggregation and state agreement of nonlinear multi-agent systems with switching topologies. *Automatica*, 45(5):1165–1175, 2009.

[33] Z. Lin, B. Francis, and M. Maggiore. State agreement for continuous-time coupled nonlinear systems. *SIAM Journal on Control and Optimization*, 46(1):288–307, 2007.

[34] H. K. Khalil. *Nonlinear systems*, volume Third Edition. Prentice hall, 2002.

[35] Nicholas J Higham. *Functions of matrices: theory and computation*. Siam, 2008.

[36] D.G. Luenberger. *Introduction to linear and nonlinear programming*, volume 28. Addison-Wesley Reading, MA, 1973.

[37] A. Forsgren and T. Odland. On the connection between the conjugate gradient method and quasi-newton methods on quadratic problems. *Computational optimization and applications*, 60(2):377–392, 2015.

[38] A. Lindquist and J. Sand. An introduction to mathematical systems theory. *KTH Lecture Notes, Division of Optimization and Systems Theory, Royal Institute of Technology (KTH)*, 1996.

[39] B. Afsari. Riemannian Lp center of mass: Existence, uniqueness and convexity. In *Proc. Amer. Math. Soc*, volume 139, pages 655–673, 2011.

[40] B. Afsari and P.S. Krishnaprasad. Some gradient based joint diagonalization methods for ica. In *Independent Component Analysis and Blind Signal Separation*, pages 437–444. Springer, 2004.

[41] A. Sarlette, R. Sepulchre, and N.E. Leonard. Autonomous rigid body attitude synchronization. *Automatica*, 45(2):572–577, 2009.